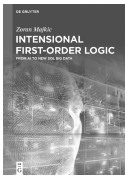
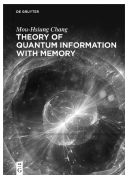
Zoran Majkić
Category Theory

Also of Interest



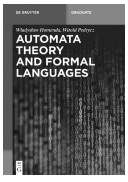
Intensional First-Order Logic
From AI to New SQL Big Data
Majkić, 2022
ISBN 978-3-11-099494-0, e-ISBN 978-3-11-098143-8



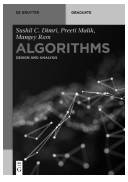
Theory of Quantum Information with Memory
Chang, 2022
ISBN 978-3-11-078799-3, e-ISBN 978-3-11-078810-5



Abstract Algebra
An Introduction with Applications
Robinson, 2022
ISBN 978-3-11-068610-4, e-ISBN 978-3-11-069116-0



Automata Theory and Formal Languages
Homenda, Pedrycz, 2022
ISBN 978-3-11-075227-4, e-ISBN 978-3-11-075230-4



Algorithms
Design and Analysis
Dimri, Malik, Ram, 2021
ISBN 978-3-11-069341-6, e-ISBN 978-3-11-069360-7

Zoran Majkić

Category Theory

Invariances and Symmetries in Computer Science

DE GRUYTER

Author
Dr. Zoran Majkić
Via Palestro 13
00185 Rome
Italy
majk.1234@yahoo.com

ISBN 978-3-11-108056-7
e-ISBN (PDF) 978-3-11-108167-0
e-ISBN (EPUB) 978-3-11-108201-1

Library of Congress Control Number: 2022949955

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available on the Internet at http://dnb.dnb.de.

© 2023 Walter de Gruyter GmbH, Berlin/Boston
Cover image: petrovval/ iStock / Getty Images Plus
Typesetting: VTeX UAB, Lithuania
Printing and binding: CPI books GmbH, Leck

www.degruyter.com

Dedicated to my daughters, Viviana and Sofia

> Dwell on the beauty of life. Watch the stars, and see yourself running with them.
>
> Meditations, Antonius Marcus Aurelius, Emperior of Rome, 121–180

> "You gods," Calipso exclaimed, "ought to be ashamed of yourselves. You are always jealous, and hate seeing a goddess take a fancy to a mortal man, and live with him in open matrimony. So when rosy-fingered Dawn made love to Orion, you precious gods were all of you furious till Diana went and killed him in Ortygia. So again when Ceres fell in love with Jason, and yielded to him in a thrice-ploughed fallow field, Jove came to hear of it before so very long and killed Jason with his thunderbolts.
> And now you are angry with me too because I have a man here. I found the poor creature sitting all alone astride of a keel, for Jove had struck his ship with lightning and sunk it in mid ocean, so that all his crew were drowned, while he himself was driven by wind and waves on to my island. I got fond of him and cherished him, and had set my heart on making him immortal, so that he should never grow old all his days."
>
> The Odyssey, book 5, by Homer

> I thought how each and every one of us is just such a tree: the taller we grow toward the sky, through the wind and rain toward God, the deeper we must sink our roots through the mud and subterranean waters toward hell.
>
> Milorad Pavić, Dictionary of the Khazars: A Lexicon Novel, 1984

Preface

> Mathematics, rightly viewed, possesses not only truth, but supreme beauty—a beauty cold and austere, like that of sculpture, without appeal to any part of our weaker nature, without the gorgeous trappings of painting or music, yet sublimely pure, and capable of a stern perfection such as only the greatest art can show. The true spirit of delight, the exaltation, the sense of being more than Man, which is the touchstone of the highest excellence, is to be found in mathematics as surely as poetry.
> Bertrand Russel, p. 60, The Study of Mathematics [1]

Most related philosophical questions deal with specific symmetries, objectivity, interpreting limits on physical theories, classification and laws of nature. The recent history of the philosophy of mathematics is largely focused on grasping and defining the nature and essence of mathematics and its objects. Attempts to do this include explicating versions of: mathematics is just logic, mathematics is just structure, mathematics is a meaningless game, mathematics is a creation of the mind, mathematics is a useful fiction, etc. For example, the basic principles of general semantics, which include time binding, are described in the publication Science and Sanity, published in 1933 by Alfred Korzybski, where he wrote these ideas:

> In general physical abstractions, including daily-life abstractions are such that—we proceed by a process of forgetting. In other words, no description or 'definition' will ever include all particulars…
> In mathematics, however, we build for ourselves a fictitious and over-simplified verbal world, with abstractions which have all particulars included. If we compare mathematics, taken as a languages, with our daily language, we see readily that in both verbal activities we are building for ourselves forms of representation for this something-going-on, which is *not* words.
> Considered as a language, mathematics appears as a language of the highest perfection, but at its lowest development…
> Our other languages would appear, then, as the other extreme, as the highest mathematics, but also at their lowest development—highest mathematics, because in them we can speak about everything; at their lowest development because they are still $\mathcal{A}$ and not based on asymmetrical relations. Between two languages there exists a yet a large unbridged structural gap. The bridging of this gap is the problem of the workers of the future. Some will work in the direction of inventing new mathematical methods and systems, bringing mathematics closer in scope and adaptability to ordinary language (for instance, the tensor calculus, the theory of groups, the theory of sets, the algebra of states and observables,…). Other will undertake linguistic researches designed to bring ordinary language closer to mathematics…
> The mathematicians who discover or invent new *methods* for relating and structures are the biggest 'mental' giants we have had, or ever shall have.
> p. 69, [2]

Symmetries play a fundamental role in physics because they are related to conservation laws. For instance, the conservation of the momentum vector is associated with translations (Poincare *transformations,* which treats time and space on an equal footing, in contrast with traditional approaches where one aims directly for a description of the "time evolution") invariance of the Lagrangian with Noether's current and conservation of the Noether's charge. Lagrangian (density) is a fundamental concept in physics,

https://doi.org/10.1515/9783111081670-201

which determines the principal property (phenomena) of the objects in physics: their time-space path derived by the variational principle from Lagrangian into the Euler–Lagrange equations. Recall that a *global symmetry* is defined as a transformation that keeps the action invariant (like Maupertious' principle in mechanics and Fermat's principle in optics).

As Professor J. B. Shaw says [3]:

> We find in the invariants of mathematics a source of objective truth. So far as the creations of the mathematician fit the objects of the nature, just so far must the inherent invariants point to objective reality. Indeed, much of the value of mathematics in its applications lies in the fact that its invariants have an objective meaning.

But what happens if we apply this method not to the objects of nature but to the mathematical abstract objects?

If we use this general pattern for the category theory, which is an *abstract mathematical theory,* we first have to find the useful analogies with that physics, and first of all the fundamental transformations (at the place of the Poincare transformations in physics) of a given category **C**. One of the fundamental transformations, provided in quasi all introductions to category theory, is the transformation of **C** into its Cartesian products

$$\overbrace{\mathbf{C} \times \cdots \times \mathbf{C}}^{n}$$

for $n \geq 2$, but *this* transformations are not characteristic for the category theory because they are only a heredity from the set-based mathematical theory where a given *set* is transformed into its Cartesian products.

Just because of that, in my second PhD thesis in Computer Science [4], I analyzed in 1994 (thanks to my previous knowledge of symmetry theory in physics, obtained when I worked as an Assistant Professor at ETF, Belgrade, in electromagnetic theory, from 1979 to 1981 and prepared my first PhD thesis), from my point of view, the most characteristic transformation of a base category **C** into the infinite hierarchy of the arrow categories (such a transformation does not exist in the set-based theory of mathematics) denominated by *n*-dimensional levels.

So, if we consider a given base category **C** as a space with a set of its properties as a "physics field theory" with specific *properties expressed by a Lagrangian*, the question is to establish, which are these different "Lagrangian" of the base category, which remain invariant under particular categorial transformations. Obviously, the more general and important properties ("Lagrangian") of a given category are its *universal* categorial properties, like its universal arrows. This intuitive idea, shortly presented in [4], here will be presented in much more details and in relationships with a number of applications in computer science, especially with the more recent developments presented in my book [5].

Recently, in [6] is presented, the Space, time and number in the brain (henceforth: STNB) searches for the foundations of mathematical thought across collection of articles, with five sections: (1) Mental magnitudes and their transformations, (2) Neural codes for space time and number, (3) Shared mechanisms for space, time and number, (4) Origins of proto-mathematical intuitions, (5) Representational change and education. This evidence provides new vistas into the area of cognitive-epistemic Foundations of Mathematics (ceFOM).

> Another reason why a nonmathematician cannot study psychological phenomena adequately is that mathematics is the only science which has *no physical content* and, therefore, when we study the performances of Smith and Brown when they mathematize, we study the *only* available working of 'pure-mind'. Moreover, mathematics is the only language which at present has a structure similar to that of the world and the nervous system... Mathematics and what is called "sublimation" in psychiatry have a similar neural mechanism, which is expressed structurally in the spiral theory, or the cyclic chain of nerve currents, where the end-product of one process becomes the starting point of the next.
> Alfred Korzybski, p. 73, p. 304, Science and Sanity [2]

Questions about how we can know numbers, shapes and rules, and how it is even possible to prove necessary arithmetical, algebraic and geometric relations; about whether mathematics is an internal cognitive structure (that evolved), a learned one and/or one existing in an independent realm; and about whether we can build a machine or a computer simulation that emulates our knowledge structures and conscious experience, are quite old.

The *identity* is usually added to the First-Order Logic (FOL) as a *primitive* logical binary predicate satisfying the properties of reflexivity, symmetry and transitivity of its arguments. It is a built-in primitive concept, which does not depend on Tarski's interpretations, different from the concept of *logic equivalence,* which is a *derivable* concept inside the logic: two sentences A and B are logically equivalent if have the same truth value for each Tarski's interpretation of a given logic theory. In an analog way, in category theory, the identity arrows are primitive arrow concepts assigned to each object in the category, and any such identity arrow is reflexive and symmetric for the source and domain object of such an arrow (if the arrow $\mathrm{id}_1 : a \to b$ is well-defined in the category, then also the arrow $\mathrm{id}_2 : b \to a$ is a well-defined arrow of this category, and if also the identity arrow $\mathrm{id}_3 : b \to c$ is well-defined then the composed arrow $\mathrm{id}_3 \circ \mathrm{id}_1 : a \to c$ is well-defined in this category and represents the transitivity property). This built-in categorial concept is not derivable by other arrows in a given category, but is a priori introduced to a given category (a model of some mathematical theory) in the moment of definition of the *objects* of this category: each object in a given category is unique, and hence identical to itself only. What we are able after this initial definition of the objects of a given category and the kind of arrows between them, is only to *recognize if two diferent object are isomorphic* objects of such a category (so, isomorphism is generally an abstract kind of equivalence different from equality): in any commutative diagram of

a given category, any particular object can be *replaced* by another object isomorphic to it. So, the isomorphism of the objects is stricltly determined by the expressive power of the arrows in a given category. Thus, the isomorphism in category theory corresponds, for example, to the logical equivalence in a given logic theory (which depends on the expressive power of logic deduction of such a logic). Consequently, both are derivable relationships *inside* their formalized theories.

Usually, in mathematics, we are using the same binary symbol $'='$ for the two fundamentally different relationships, without an explicit distinction between a primitive built-in identity concept and derived equality concept. The difference between the primitive concept of *identity* and derived concepts of various kinds of equivalences can be easily shown in the algebra of natural numbers: the statement $1 = 1$ expresses the primitive (self) identity concept of the number 1, while the statement $x^2+2xy+y^2 = (x+y)^2$ is a derivable relationship of this algebra based on two operations of addition and multiplication. We can verify this derivable equality relationship, inside this theory, by assigning the natural numbers to the variables x and y. For example, value 1 to x and value 2 to y, so that the equation above *reduces to identity* $9 = 9$. That is, *we need* the built-in relationship of identity *inside* this mathematical theory in order to derive more complex relation of equality of algebraic terms with variables.

Analogously, the isomorphism of two objects in a given category is demonstrated by using the built-in identity arrow concepts of a given category. The famous statement "*up to isomorphism*" in category theory just provides explicitly this fact: what we can derive inside any well-defined category are (only) the isomorphisms (substitutivity of the objects) of its objects, but *not the identity* of the objects. The isomorphism is a more abstract equivalence relationship than the primitive identity, so that two nonidentical objects can be isomorphic (nonrecognizable inside this category as a different object, and hence one can be substituted by another isomorphic (but different) to it): in this case, the "abstraction" means that we forget some specific internal properties of the objects (they remain hidden to the expressive power of the arrows).

However, independently, how complex is the internal structure of a given object a of a given category, the identity arrow does not give any other useful information about this internal structure. An identity arrow only tells us that the domain an codomain objects of this arrow are equal, so that such an arrow in any diagram of arrow compositions of this category can be eliminated. In effect, in the category theory usually we avoid to explicitly represent the identity arrows, and use them only in rare and particular cases. Note that any property of the objects (of their internal structure) are defined by particular commutative diagrams in which this object participates as a node of this diagram, usually with universal arrows as a part of such commutative diagrams. So, it holds that the representation of the internal, possibly complex, structure of an object depends on the expressive power of the arrows of a category: usually, such arrows can show only a part of the structure of complex objects, while the rest remains hidden. That is, in effect, the reason that the category theory is fundamentally abstract. So, by choosing the

expressive power of the arrows in a given category, we establish the level of abstraction for its objects (the level of "observable" properties of the object in this category). The significant example of this abstraction is provided here in Chapter 7, dedicated to conceptual symmetry of the database **DB** category.

It is similar, generally, to the object-oriented databases, where the objects can be very complex and within reach to internal structures, but we define only the methods for this object (representable by the arrow in a particular category) that show a part of the information encapsulated in this object to other objects (the codomain of such arrow-methods) in an object-oriented application. Thus, we cannot distinguish, in such an object-oriented application, two different objects, which offer the same observations of their internal structures for a given set of implemented methods (set of arrows in a category expressing this object-oriented application). Thus, in such an object-oriented application, we have the *observationally-equivalent* objects also if they have different hidden parts of their internal structure. An example for these observational equivalences is provided in Section 7.2 (and its subsections) of Chapter 7. This phenomena of "observationally-equivalent" objects in a given category is expressed by the categorial *notion of isomorphism*. However, we are able to define different observational equivalences (strong, weak, etc.), but only one of them corresponds to the isomorphism in a given category.

To show an isomorphism of two objects a and b of a category, we need to explicitly use their identity arrows, id_a and id_b, relatively: in order to specify that these objects are isomorphic (so that one *can be replaced* by other one), denoted by $a \simeq b$, we must have two arrows $f : a \to b$ and $g : b \to a$, such that $f \circ g = \mathrm{id}_b$ and $g \circ f = \mathrm{id}_a$. So, the identity arrows are necessary in order to establish that two objects are *abstractly equal* (or observationally equal in a category with fixed expressive power of its arrows), or *isomorphic*, i. e., "equal *up to isomorphism.*" Consequently, also if two objects a and b are really equal (but we defined them by two different labels), *we are not able* to discover (by the categorial means) that they are equal really, but only that they are isomorphic. In effect, by any declaration of some object in a category by its label (concrete symbol or name), we automatically insert its identity arrow in this category as well, and there is not any other way to derive identity arrows different from the composition of already defined equal arrows (by the transitive property of identity arrows).

However, also this observation does not legitimate us to identify any concrete object in a category by its identity arrow, from the fact that such identity arrow does not tell anything about real internal structure of this object.

Moreover, we have the following categorial asymmetry: in any category, for each its object there exists also its identity arrow, while generally for each arrow in the category we have not an object of the same category representing this arrow at least "up to observational equivalence." Only in some particular category, like **Set** (with small sets as objects and small functions as arrows), each function $f : a \to b$ from the small set a into the small set b, can be "up to observational equivalence" represented by the small set (thus an object), $c = \{(x, f(x)) \mid x \in a\}$, that is with the *graph* of this function. This

term "*up to observational equivalence*" used for the representation of the arrow (function) f by the object (set) c means that they *are not equal*: in effect, in the codomain b of the arrow (function) f there are also the elements, which are not present in the set (object) c.

This observation was one of the start points for the investigation of the *categorial symmetry* in my PhD work.[1] The initial motivation of transformation of the arrows in the representative objects was connected with *deductive systems as* categories (not necessarily the preorder categories) where the objects are logic sentences and the arrows between them are the deduction paths (derivations) from a given (source) sentence A to effectively derived sentence B (we accept the number of different deduction paths from A into B as well). For example, in constructive mathematics (as intuitionistic logic, for example), the sentence $A \Rightarrow B$ is true if there is an effective deductive process of derivation of sentence B from the sentence A, represented by an arrow $f : A \to B$ in this category. The relative problem, of how to define *functorially* the implication for such constructive logic from the effective deductions (arrows in such a category), will be presented in this book as well.

What is important in such functorial method, for derivation of the objects in a given category $\mathbf{C}$ from its arrows, is that it *needs* to use necessarily the arrow categories $\mathbf{C} \downarrow \mathbf{C}$ where the arrows $g : a \to b$ of $\mathbf{C}$ are encapsulated as the objects in $\mathbf{C} \downarrow \mathbf{C}$ denoted by $J(g)$. Thus, the functorial representation has to be provided by a functor $F : \mathbf{C} \downarrow \mathbf{C} \to \mathbf{C}$, and hence, for each arrow $g : a \to b$ of $\mathbf{C}$, the object $F(J(g))$ of $\mathbf{C}$ has to be, "up to observational equivalence," the representation of the arrow g *in the same* category $\mathbf{C}$. With this method, we introduce the infinite hierarchy of the arrow categories, for a given base category $\mathbf{C}$, as follows:

The notion of hierarchy of the arrow categories, for a given base category $\mathbf{C}$, is defined inductively for $n \geq 1$ by

$$(\mathbf{C} \downarrow \mathbf{C})^1 \triangleq \mathbf{C} \downarrow \mathbf{C}, \quad (\mathbf{C} \downarrow \mathbf{C})^{n+1} \triangleq (\mathbf{C} \downarrow \mathbf{C})^n \downarrow (\mathbf{C} \downarrow \mathbf{C})^n \tag{1}$$

and, derived from it, the *n-dimensional levels* based *only* on the base category $\mathbf{C}$, instead on the its first arrow category $(\mathbf{C} \downarrow \mathbf{C})$ used in (1) for $n \geq 1$, denoted by $\mathbf{C}_{n+1} \triangleq (\mathbf{C} \downarrow \mathbf{C})^n$. That is, the n-dimensional levels are defined inductively, for $n \geq 1$, by

$$\mathbf{C}_1 \triangleq \mathbf{C}, \quad \mathbf{C}_{n+1} \triangleq \mathbf{C}_n \downarrow \mathbf{C}_n \tag{2}$$

i. e., inductively, $(n + 1)$-dimensional level is the arrow category obtained from the n-dimensional level, where the arrows of the n-dimensional level are composed by 2^{n-1}

1 This kind of internal symmetry is applied in my book for data integration theory [5] and recently in my book of Intensional First-order Logic [7] (in Section 1.3.2 dedicated to "logics via symmetry") applied to the conservative (invariant) logic knowledge transformations as the intensional abstraction transformations; moreover, I considered ontological encapsulation transformations as well.

arrows of the base category **C** (considered as the first 1-dimensional level as well, which represents the leafs of the *binary tree* of this *n*-dimensional syntax structure).

Remark. Note that such an infinitary hierarchy of *n*-dimensional levels there exist for any category. So, it holds also for the minimal (nonempty) discrete category **1** composed by one unique object and unique identity arrow of this object. This infinitary *n*-dimensional levels of the base category **1** there exist just on the compositionality of the identity arrow with itself, and show to us how also such atomic (minimal) category generates the *infinite* hierarchy of the arrow categories derived from **1**. The existence of the identity arrows of the objects, in *any* other finite or infinite category, guarantees the existence of the infinitary *n*-dimensional hierarchy for each category as well. □

So, this phenomena is a *foundational* and characteristic (distinguishable) property of each category, and hence *of the category theory*, and similar to the foundational role of sets in the traditional set-based mathematics, or to the role of primitive built-in binary identity predicate in the first-order logic. In what follows, it will be used for the new multidimensional categorial definition of the set of natural numbers $\mathbb{N}$, different from the standard von Neumann set-based flat bidimensional definition.[2]

Thus, we can define inductively the set of natural numbers in a categorial way, based on the base category $\mathbf{C} = \mathbf{1}$ (and hence in a nonset-based way, as in von Neumann inductive definition), in the following way: the number zero is defined by empty category **0**, while others are defined inductively from the base category **1**, i. e.,

$$0 = \mathbf{0}$$
$$1 = \mathbf{1}$$
$$2 = \mathbf{1} \downarrow \mathbf{1}$$
$$3 = (\mathbf{1} \downarrow \mathbf{1}) \downarrow (\mathbf{1} \downarrow \mathbf{1})$$
$$4 = ((\mathbf{1} \downarrow \mathbf{1}) \downarrow (\mathbf{1} \downarrow \mathbf{1})) \downarrow ((\mathbf{1} \downarrow \mathbf{1}) \downarrow (\mathbf{1} \downarrow \mathbf{1})),$$
$$\text{etc.}$$

The fact that this inductive definition is not set-based, but an *n*-dimensional geometric definition, it is easy to verify by the consideration of what geometrically represent the

2 In defining the natural numbers, von Neumann begins by examining the most fundamental set, the empty set denoted by {} and created the following inductive pattern for the set-based definition of natural numbers:

– number zero, 0 = {} has zero elements;
– number one, 1 = {{}} has one element (the empty set);
– number two, 2 = {{}, {{}}} has two elements (the empty set and the set containing the empty set);

and this process would continue infinitely until all the natural numbers have been defined. Note that in this set-based definition of natural numbers is introduced the "member of" relation $\in$, e. g., $0 \in 1$, $0 \in 2$ and $1 \in 2$, etc.

arrows in the hierarchy of the arrow-categories that define the natural numbers. More intuitive geometrical/algebraic representation of the pattern above, used for definition of natural numbers, will be provided in first Section 1.1.

In this book, we will investigate two kinds of the transformations with mathematical symmetries in the category theory [4]:

1. The first one, provided in Chapter 2, regards the *invariant* categorial properties (adjunctions, (co)limits) under comma-propagation *transformations* from a given n-dimensional level category $\mathbf{C}_n$ into $(n+1)$-dimensional level, representing a *global symmetry*.
2. The second one, provided in Chapter 3, regards the *transformation* of the arrows into conceptualized objects. In most cases, they are descending functorial transformations (by a closure functor T_e) from a $(n + 1)$-dimensional level into n-dimensional level: the process of conceptualization of the arrows into objects of a given n-dimensional level. This kind of *internal symmetry* regards the relationship between two primitive concepts of the category theory: objects and arrows.

Symmetries play a fundamental role in physics because they are related to conservation laws. This is stated in Noether's theorem, which says that invariance of the action under a symmetry transformation implies the existence of a conserved quantity. Albert Einstein once said, about "*the world of our sense experiences,*" and "*the fact that it is comprehensible is a miracle*" (1936, p. 351). A few decades later, another physicist, Eugene Wigner, wondered about the unreasonable effectiveness of mathematics in the natural sciences, concluding that "*the miracle of the appropriateness of the language of mathematics for the formulation of the laws of physics is a wonderful gift, which we neither understand nor deserve*" (1960, p. 14). At least three factors are involved in Einstein"s and Wigner"s miracles: the physical world, mathematics and human cognition.

It is interesting fact that both of them considered the standard quantum mechanics (QM) as an *incomplete* and statistical theory, so that for the description of the individual particles we needed a new more sophisticated theory with hidden variables. This was a starting point for me to develop this new and complementary part of QM from 2010 to 2019 published in three volumes [8–10] for completion of QM and unification with Einstein's General Relativity.[3]

3 This work in some way compensated my nonfinished first PhD thesis, when I received a formal invitation from the Department of Electromagnetic Theory from Professors Gerosa and Bardati: they offered me a very modern laboratory for the experimentation of microwave radiation, a new method for anticancer therapy in 1978. However, this study has been interrupted in 1981 by a decision of the faculty committee at ETF (Nikola Tesla), Beograd, when I worked as a young Assistant Professor at the Microwave Department. They did not give me the 6 months nonpay permission to finish my PhD at La Sapienza University at Rome, Italy.
Obviously, such a decision was not based on academic considerations of my scientific work and preparation, because I obtained my Assistant Professor position, and was accepted positively by *all* professors

The approach to the *analogy* relationship between physics and category theories assumed can be summarized as "adjunction-as-field" in the following table:

Physics Theory	**Category Theory**
Continuous time space with points p_i	Category **C** as a flat space with objects p_i as points and arrows as directed paths between them
Scalar field Ψ of a given type **J**	Adjunction $(\triangle, G, \varepsilon, \eta)$ with diagonal functor $\triangle : \mathbf{C} \to \mathbf{C}^{\mathbf{J}}$ of type defined by small index category **J**
Lagrangian density $\mathcal{L}(\Psi, p_i)$	Universal arrow $(\triangle(p_i), \eta(p_i))$ at a point p_i
Action $S = \int \Omega_g \mathcal{L}(\Psi, p_i)$	Set of displacements of each point p_i (path integrals) represented by arrows with domain objects at p_i
Principle of *minimal* action	Property that for each cone of a diagram d' in $\mathbf{C}^{\mathbf{J}}$ there is a *unique* arrow to the limit cone
Euler–Lagrange equation with position variable p_i of a material object	Commutative diagram in **C** of the adjunction for a fixed diagram d' ("material object") in $\mathbf{C}^{\mathbf{J}}$ and for each position variable p_i

This interpretation of categorial symmetry, based on the comma-propagation transformations is discussed in detail in Section 2.5.

of the ETF. The leading professors at my faculty, from another department of electromagnetic theory, wanted to take under their domain my department as well, by substituting Professor Milanović (I accepted his invitation to work as his assistant, and after 2 years I worked in the Institute VTI where I finished my MSc in microwaves) by some people from their department (nonspecialized in microwaves). By my fast completion of the PhD in only 6 months in Italy, I would obtain the position of Associated Professor at my ETF department, and hence impede their plan. However, with their strong, institutional and political relationships inside the faculty, the faculty committee (Naučno Veće) rejected the permission for me to be absent from the faculty for next 6 months (contrary to the practice of permission for 3 years necessary for PhD studies at foreign universities). I understood that with this fact my academic carrier was finished, that I would not be able to remain and continuously suffer this injustice. So, I interrupted my PhD thesis and my faculty position, and I left definitely my country Jugoslavia.
I obtained the second opportunity for PhD studies 15 years after the interruption of my first PhD. In practice, with my already realized software, the material for my PhD in knowledge representation and forward-chaining expert systems inference was ready. I presented it shortly to Professor M. Lenzerini (who worked with semantic networks and frame-based models in the same period at La Sapienza) as my program for the eventual PhD thesis, and after the positive exam for the selection of PhD students in computer science. However, after the seminar of Giuseppe Longo in a categorial model for typed lambda calculus [11], I started with completely new ideas for my thesis, and completed it after 4 years in 1998.

Dependencies between the chapters

This book considers two kinds of symmetry. In Part I, included in Chapters 1 and 2, is dedicated to the development of abstract category theory for the arrow categories and *global symmetries* of comma-propagation transformations. Chapter 1 is an introduction of the infinite hierarchy of the arrow categories considered as "*Galilean reference frames*" of a given base category $\mathbf{C}$ (the n-dimensional levels $\mathbf{C}_n$, for $n \geq 1$), the two fundamental comma projection-functors with natural transformation ψ between them and inverse to it is an encapsulation operator J. We provide a new categorial-based (nonset-based) definition of natural numbers. Then we introduce the basic transformation of functors and natural transformations along this sequence of n-dimensional levels, from n-dimensional to $(n + 1)$-dimensional level, denominated as "comma lifting," and the "comma-induction" (the consecutive iteration of comma lifting) with particular application to the adjunctions of functors.

Chapter 2 is dedicated to the more general phenomena of *comma-propagation transformations* (a "Galilean boost" of "uniform motion" between two n-dimensional levels seen as different reference frames) and its invariant properties. Comma-propagation is a general transformation between n-dimensional levels, applied to the functors and natural transformations. If we consider the universal properties of a base category (universal arrows, adjunctions) as a kind of different "Lagrangian" in the category theory, then we will consider their invariance under this comma-propagation transformation: a *global symmetry* in this case means that these categorial universal structures are preserved under these general transformations *in all* n-dimensional levels (the law of conservation). The application of this kind of symmetry is provided as well in dedicated sections of this chapter.

Part II from Chapter 3 to Chapter 7, is dedicated to another fundamental, categorial *internal symmetry* between two primitive categorial concepts, objects and arrows. Chapter 3 is theoretical definition of this internal symmetry and categorial invariance, relative to the *transformation of arrows into the objects*, and can be done only for the categories, which satisfy particular conditions (the transformation of objects into arrows of a given category is a standard one: transforming a given object into its identity arrow, and is valid for all categories). It provides a particular hierarchy for the quality of the symmetry between arrow and objects and the relative properties of the categories that must be satisfied in each level of hierarchy. Other chapters are dedicated to a number of applications of this symmetry in a complex-incremental way: Chapter 4 is dedicated to logical deduction (to natural deduction and propositional intuitionistic logic), Chapter 5 to lambda calculus, Chapter 6 to the Labeled Transition Systems (LTS), and the final Chapter 7 to data integration theory. Additional material for these applications is provided in the Appendix A.

https://doi.org/10.1515/9783111081670-202

Detailed plan

Chapter 1: In Section 1.1, after the introduction of the n-dimensional levels (the hierarchy of the arrow categories for a given base category $\mathbf{C}$), we define two basic operations: the natural transformation ψ between first and second comma projection functors $F_{\mathrm{st}}, S_{\mathrm{nd}} : \mathbf{C}_{n+1} \to \mathbf{C}_n$, which defines arrows from the objects, and its inverse encapsulation operation J. Then we introduce a new categorial definition of natural numbers. Section 1.2 introduces the "comma lifting" of categories, functors and natural transformations from a n-dimensional level into next $(n+1)$-dimensional level, and diagonal functor $\blacktriangle : \mathbf{C}_n \to \mathbf{C}_{n+1}$. Then we show that, for each n-dimensional level $\mathbf{C}_n$, $n \geq 2$, and its identity endofunctor $\mathrm{Id}_{\mathbf{C}_n}$, there exist two natural transformations $\sigma : \blacktriangle \circ F_{\mathrm{st}} \xrightarrow{\cdot} \mathrm{Id}_{\mathbf{C}_n}$ and $\sigma^{-1} : \mathrm{Id}_{\mathbf{C}_n} \xrightarrow{\cdot} \blacktriangle \circ S_{\mathrm{nd}}$, such that their vertical composition satisfies $\sigma^{-1} \bullet \sigma = (\psi; \psi)$. Based on this comma lifting, in Section 1.3 we provide the comma-induction along the n-dimensional levels hierarchy, in four different possible cases, for the functors and natural transformations. The distributive low for the comma-induction w.r.t the composition of functors, horizontal and vertical composition of natural transformations and the property of closure for the strict comma-induction is shown. Finally, in Section 1.4 we prove that the comma-induction preserves the adjunctions between the functors.

Chapter 2: In the previous chapter, we only considered the pure arrow categories (n-dimensional levels hierarchy) in order to show the basic properties of the comma-induction (the propagation of functors and natural transformations from base categories to all higher n-dimensional levels). Here, in this chapter, we will introduce also the functor categories $\mathbf{C}_n^{\mathbf{J}}$, where $\mathbf{J}$ is *any* small index category. In this way, we can consider the propagation of the (co)limits (from a base category) inside the infinite hierarchy of n-dimensional levels. Consequently, in Section 2.1 we define the comma-propagation as a generalization of the comma-inductions provided in Chapter 1: comma-induction is a specific case of this more general comma-propagation (the case when the index category $\mathbf{J} = \mathbf{2}$ is the preorder category with two objects 0 and 1). So, we introduce also a generalized (w. r. t. the standard) diagonal functor $\triangle : \mathbf{D} \to \mathbf{C}^{\mathbf{J}}$, and two auxiliary functors, $L : (\mathbf{C}_n^{\mathbf{J}} \downarrow \mathbf{C}_n^{\mathbf{J}}) \to (\mathbf{C}_n \downarrow \mathbf{C}_n)^{\mathbf{J}}$ and $K : (\mathbf{C}_n \downarrow \mathbf{C}_n)^{\mathbf{J}} \to (\mathbf{C}_n^{\mathbf{J}} \downarrow \mathbf{C}_n^{\mathbf{J}})$, such that their compositions $K \circ L$ and $L \circ K$ are equal to identity endofunctors. Based on these two auxiliary functors, in Section 2.2 we define the comma-propagation transformations as a generalization of comma-induction provided in Chapter 1. Consequently, in Section 2.3 we consider the comma-propagation of the adjunctions and (co)limits in all infinite hierarchy of n-dimensional levels, and we provide some examples (for generalized diagonal functor). The examples of comma-propagation of the (co)products and equalizers are provided in Section 2.4 of this chapter. Finally, in Section 2.5, we analyze the *global symmetries* for the categorial "Lagrangians," i. e., universal structures like universal arrows and adjunctions, w.r.t to the comma-propagation transformations, and in relationship with the Galilean boosts in physics.

https://doi.org/10.1515/9783111081670-203

Chapter 3: In this chapter, we introduce a kind of *internal symmetry* based on the invariant transformation of the arrows into the "conceptualized" objects. This invariance means that in the categories able for such transformations, this transformation does modify them. Each object obtained as a transformation of some arrow of the category is "equivalent" to this arrow (in an analog way in which each object of a category can be represented by its identity arrow), and the composition of the arrows in this category can be equivalently represented by analog composition of their conceptualized objects. Thus, we consider the categories for which there is this symmetry between primitive categorial concepts: arrows and objects. Then, in Section 3.1, we introduce also a quality-hierarchy of such internal symmetry for the categories. In this section, we provide also a simple example: A category for predicate logic with intensional abstraction. After this introduction, in Section 3.2 we define the metacategory for such perfectly symmetric categories, provided with the operator $B_{\top}$ for transformation of the arrows into the conceptualized objects and the new operation "$*$" for composition of these conceptualized objects, and we also present some examples for the two lowest, quality-level internal symmetry. Then, in Section 3.3, we define the 3rd symmetry quality-level, for the so-called Conceptually Closed Categories (CoCC) when there exist a symmetry closure functor $T_e : \mathbf{C} \downarrow \mathbf{C} \to \mathbf{C}$, with the natural isomorphism $\varphi : T_e \circ \blacktriangle \simeq I_{\mathbf{C}}$ (when the transformation of the identity arrow of an object is equal up to the isomorphism to this object). We show that the category **Set** is an example for this symmetry quality level. We prove that, for each CoCC category, there exist two natural transformations $\tau : F_{\mathrm{st}} \xrightarrow{\cdot} T_e$ and $\tau^{-1} : T_e \xrightarrow{\cdot} S_{\mathrm{nd}}$, defined by $\tau = (T_e\sigma) \bullet (\varphi^{-1}F_{\mathrm{st}})$ and $\tau^{-1} = (\varphi S_{\mathrm{nd}}) \bullet (T_e\sigma^{-1})$. Next, Section 3.4 is dedicated to the higher quality level of categorial internal symmetry: the symmetry extended categories, for which is satisfied the vertical composition property $\tau^{-1} \bullet \tau = \psi : F_{\mathrm{st}} \xrightarrow{\cdot} S_{\mathrm{nd}}$. In such categories, each arrow $f : a \to b$ is split into the composition $\tau^{-1}(J(f)) \circ \tau(J(f))$ of two arrows $\tau(J(f)) : a \to \widetilde{f}$ and $\tau^{-1}(J(f)) : \widetilde{f} \to b$, where $\widetilde{f}$ is the conceptualized object obtained from the arrow f. We prove that this symmetry property is an invariant w. r. t. comma-induction transformation, i. e., and it is valid in all higher n-dimensional levels. Finally, in Section 3.5, we consider the upper limit for this symmetry property: the imploded categories, when $\blacktriangle \circ T_e \simeq I_{\mathbf{C}\downarrow\mathbf{C}}$ as well. In this extremal categorial symmetry case, we obtain that all n-dimensional levels are equivalent to their base category. That is, all infinite hierarchies of n-dimensional levels are flattened inside their base category.

Chapter 4: This chapter provides an example of conceptual (internal) categorial symmetry applied to the propositional logics. In Section 4.1, we introduce the natural deduction of the positive fragment of propositional logic (without a negation logical operation), with two different tagging techniques: the Gentzen/Prawitz deduction trees and the sequent-based deduction method. Both of these are traduced into a categorial framework by the definition of the category **ND**. In Section 4.2, we demonstrate that each category **ND**, given an initial set of assumptions Γ, is a symmetry-extended category by defining the closure-symmetry functor $T_e : \mathbf{ND} \downarrow \mathbf{ND} \to \mathbf{ND}$. Then we show that the log-

ical implication has the constructive I. E. Brower interpretation, represented by the covariant functor T_e. The properties of this covariant implication functor are provided in Section 4.2.1 and we show that **ND** is not a Cartesian Closed Category (CCC), but there is a partial adjunction (p-adjunction) between T_e and the categorial product functor (shown in Section 4.2.2) so that **ND** is a partial CCC (pCCC) with a maximal p-adjunction and corresponding exponentiation diagrams. Then, in Section 4.2.3 we prove that such categorial property is preserved under comma-induction transformations to all n-dimensional levels of the base category **ND**. In Section 4.3, we use the Gentzen LJ Sequent System, with the negation logical operation as well, and hence we extend **ND** into new propositional intuitionistic logic category **IC** provided also by the symmetric tensor product $\otimes$ for categorial interpretation of the structural "comma" syntax operation used in the sequents, and two new constraints for the inference rules dedicated to the negation logical operation.

Chapter 5: This chapter provides an example of conceptual categorial symmetry applied to the lambda calculus. In Section 5.1, we introduce the proposition-as-types interpretation of the expressions $M : a$ of typed lambda calculus, where "a" is a type (a proposition) of a term M (a proof of this proposition "a"), with the set of equational axiom schemas (rewriting rules). It is well known that a categorial model is a Categorial Closed Category (CCC) with products and exponentiation objects for types and lambda terms as its arrows. Then we consider how an intuitionistic sequent in Section 4.3 can be used for the lambda calculus, and the fixed-point operator extensions for the untyped lambda calculus. With such an extension, we can simulate recursive definitions, and hence obtain the full computational power for such a programming language. So, in Section 5.2 we introduce the reflexive objects in the CCC for a categorial interpretation of fixed-point operators. Then we show how we can use a CCC category **C** provided by the categorial symmetry (CoCC) with the symmetry-closure functor $T_e : \mathbf{C} \downarrow \mathbf{C} \to \mathbf{C}$, in order to define for each its reflexive object b a subcategory $\mathbf{ER}_b$ of idempotents on b. Such a category $\mathbf{ER}_b$ of idempotents on b is used in the next Section 5.3 to support the fixed-point operators. In fact, we show that this conceptually closed category $\mathbf{ER}_b$ by the symmetry-closure functor T_e is a CCC as well.

In Section 5.4, we demonstrate that each idempotent object of $\mathbf{ER}_b$, $\widetilde{f} = T_e(J(f))$, where f is an idempotent arrow (a term of untyped lambda calculus) in a CCC category **C**, has a fixed-point operator. Thus, a CCC category **C** with reflexive objects, having categorial symmetry (CoCC), is a model for untyped lambda calculus with fixed-point operators. The categorial symmetry permits to distinguish a program (an arrow f of **C**) viewed as such, from the same program *viewed as data* (a conceptualized object $\widetilde{f} = T_e(J(f))$ of **C**).

Finally, in Section 5.5 we generalize the application of internal symmetry to idempotent completion, topological K-theory and the universal property of the Karoubi envelope.

Chapter 6: This chapter provides an example of conceptual categorial symmetry applied to the Labeled Transition Systems (LTS). In Section 6.1, we introduce the transition

systems (directed graphs) and their bisimulations. We provide an example of nondeterministic mobile processes as well. In Section 6.2, we consider the finite trees obtained by unfolding LTS. These finite trees are based on the meet-semilattice monoid of labels A^* ordered by prefix, which defines a locally posetal 2-category **A**, and hence we show that it is just a symmetry-extended category. So, we introduce the finite labeled trees (FLT) as a set X of paths in which extensions are the stings in A^*. Then, in Section 6.3 we define a new category **Z** with the objects FLTs, and arrows betweem them such that increment the agreement between the paths of FLTs (for each arrow, its target FLT object has better agreement (of proper paths) than its source FLT object), and we show that it is a symmetry-extended and locally posetal 2-category such that the 2-cells are 1-cells as well. We define the coproducts (a nondeterministic gluing the roots of two LFTs) and tensorial products in this symmetry-extended category **Z**.

The second part of this chapter is dedicated to the strong bisimulations between FLTs. So, in Section 6.4 we introduce the category of relations **Rel** and its n-dimensional levels, in order to be able to represent the transition systems and strong bisimulation as the greatest fixed points in the n-dimensional level $\mathbf{Rel}_3$, and show that **Rel** is a symmetry-extended category with the pullback square diagrams. In the next Section 6.5, we show that each transition system, represented by a finite tree M with nodes labeled by a monoid N^* of natural numbers (a tree $M \subset N^*$ is a prefix closed subset), defines a pullback square diagram in **Rel**, which is an object in $\mathbf{Rel}_3$, and hence in $\mathbf{Rel}_3$ we can represent this transition system as a fixed-point object. Finally, in Section 6.6 we also define the strong bisimulation as the greatest fixed point in the n-dimensional level $\mathbf{Rel}_3$. In this proof, we use the coinductive definition of bisimilarity rules, and we discuss this bisimulation in the context of modal logic as well. The next two subsections are provided as interesting examples of logics for the application of bisimulation, by considering the mathematical concept of symmetry (knowledge invariance) for the transformation of many valued into 2-valued logic programs. We provide a new method of ontological encapsulation of a many-valued logic (e. g., based on Belnap's bilattice) into 2-valued modal logic using semantic reflection transformations.

Chapter 7: This chapter provides an example of conceptual categorial symmetry applied to the data integration theory, based on the DB theory given in the Appendix A, Section A.3. In Section 7.1, we introduce the category **DB** [5] with the database's instances as object and morphisms, which are atomic and complex arrows f with their semantic based on the *information flux* $\widetilde{f} = B_{\top}(f)$ where $B_{\top}$ is the duality operator of internal symmetry, which transforms arrows (DB mappings) into objects (databases composed by all views transferred from the domain into codomain database). The identity of two arrows is defined by equality of their information fluxes. In next three subsections, we define canonical form and show the properties of the morphisms of this category and that **DB** is not a topos. Then we define the powerview endofunctor and monad $T : \mathbf{DB} \to \mathbf{DB}$, and demonstrate the duality property of the **DB** category. In Section 7.2, we consider the properties of the objects of the **DB** category, database federation and separation opera-

tors for the objects of **DB**, and the strong (behavioral) and weak observational equivalences between the objects (databases) of **DB**.

The principal part of this chapter is presented in Section 7.3, dedicated to the internal symmetry of **DB** with a fundamental theorem, which demonstrates that **DB** is a symmetry-extended category. In Subsection 7.3.1, we introduce the (co)products, while in Subsection 7.3.2 it is extended to all (co)limits and to exponentiation, and we show that **DB** is not a CCC. The final Subsection 7.3.3 is dedicated to Kleisli semantics, by considering the monad T (introduced in Section 7.1) as denotational semantics for database mappings. It is shown that the Kleisly category $\mathbf{DB}_T$ is isomorphic to **DB** and can be "internalized" in **DB** by the faithful and forgetful functor. In Section 7.4, we define the partial ordering $\preceq$ for objects (databases) of **DB**, extended to morphisms (based on categorial symmetry) as well: 2-cells of the 2-category **DB** are elements of this partial ordering. Each 2-cell is represented in **DB** as an ordinary monomorphism (1-cell). In Subsection 7.4.1, we introduce the matching tensor product $\otimes : \mathbf{DB} \times \mathbf{DB} \to \mathbf{DB}$ and we show that **DB** is a strictly symmetric idempotent monoidal category. The merging operator is defined for each database (object) A by endofunctor $A \oplus _ : \mathbf{DB} \to \mathbf{DB}$ in Subsection 7.4.2, while in Subsection 7.4.3, we provide the universal algebraic properties of the **DB** category. The matching and merging operators are meet and join operators of the database lattice w.r.t partial database ordering $\preceq$. So, in Subsection 7.4.4, we introduce the complete lattice L_{DB} for the partial ordering $\preceq$ of the objects (databases) of **DB** with bottom $\perp_0$ and top Υ objects. This complete lattice is just a subcategory $\mathbf{DB}_I$ of **DB**, composed by only monomorpisms (partial ordering $\preceq$), and we show that $\mathbf{DB}_I$ is an algebraic lattice of databases. Finally, we show that **DB** is concrete, locally small and a locally, finitely presentable category. The top object Υ, which is the matching monoidal unit, is finitely presentable as well.

Finally, in Section 7.5 we define the hom-objects in **DB** and we demonstrate that **DB** with matching tensor product $\otimes$ is an enriched *monoidal closed* category. So, in Subsection 7.5.1 we show that **DB** is a V-category enriched over itself: the category **DB** is an object of V-cat and the powerview endofunctor (enriched monad) $T : \mathbf{DB} \to \mathbf{DB}$ is an arrow in V-cat. In Subsection 7.5.2, instead, we provide an internalized Yoneda embedding to the **DB** category, by the contravariant functor $\overline{H} : \mathbf{DB}^{\mathrm{OP}} \to \mathbf{DB}^{\mathbf{DB}}$.

Acknowledgments and short history of this book

This book is an extension of the basic work published in my second PhD thesis from 1994 to 1998 at "La Sapienza," Roma. I worked with a number of good professors and research in logic-based AI, and especially in the research group of my PhD advisor, Professor M. Lenzerini, who in that period was the chief of the PhD programs and research in knowledge basis and AI. However, another department of mathematics organized a seminar (a graduate course) for Professor Giuseppe Longo, Director of Ecole Normal at Paris, for his recent book in category theory [11]. I was one of the rare PhD students from computer science to follow this optional seminar, because a major part has been constituted by professors and senior post-doc research in mathematics. My background in category theory was at a very abstract level [12], by studding it by myself both with universal algebra in the second part of the 1980s, so I could follow it with good understanding. This course was prevalently based on the significative applications of the category theory in various fields of computer science (as CCC models of typed lambda calculus, for example). This was probably the reason that Giuseppe noted my presence and, at the end of his seminar, offered me his book with personal dedication.

This event completely changed my original PhD program. I abandoned all of my previously developed work on my thesis and decided to begin this new scientific adventure for my new thesis in pure and applied category theory. I informed my adviser, Lenzerini, about this my decision, but at this point I remained alone without any help from my department, because not one of them had the specific knowledge in this abstract mathematics field. I warmly thank Giuseppe Longo, who immediately confirmed the originality of my mathematical work and was accepted to become my second advisor after the first version of my PhD thesis at the end of 3 years in the standard PhD program. Longo also invited Professor Sergei Soloviev to verify the mathematical correctness of my first version. In this first version, Sergei found one erroneous application of this new developed category theory, but with his help, I obtained one more year in order to improve, extend and finish my thesis.

Note that during these 4 years that were dedicated to the PhD exams and development my original thesis, I was obligated to work full-time as a free-lance IT consultant for IBM, in order to economically support my family. So, after the official presentation of my thesis at the University of Torino, and the deposition of my thesis in the national Florence University Library, I was very tired and did not have the possibility for dissemination of the obtained results to the scientific community, nor the possibility to present it in the appropriate conferences. Moreover, I did not have enough experience for Latex, and hence my PhD thesis was written (with difficulty) with MS Word. After a number of years, I also lost the portable disc with original electronic copy of my PhD.

I returned again to Lenzerini's department in 2002 as a post-doc researcher, 4 years after my PhD, to work for the European interuniversity project of Semantic Web, called the SEWASIE project IST-2001-34825. During that period, Lenzerini was the chief research leader in data integration, and it was also the very beginning of P2P data

https://doi.org/10.1515/9783111081670-204

integration systems, based on first-order logic and its second-order extensions for the interdatabase mappings. All of my work in this project was just in this working logic framework, but I tried to consider many-valued logics to overcome intricate problems for the inconsistences and to find a more robust data integration P2P system, trying to overcome the strong (extensional) mapping between databases by using intensional first-order logic [7]. With support of the department, I published a dozen papers, and I developed clear ideas about the development of a new **DB** category for semantic mappings between databases. I warmly thank Eugenio Moggi and Giuseppe Rosolini for their invitation to a seminar at DISI Computer Science, University of Genova, Italy, December 2003, and for useful discussion that has offered me the opportunity to make some corrections and to improve an earlier version of that research.

However, no one in the department followed my ideas, and after an invitation by the Director V. S. Subrahmaninan at the UMIACS Laboratory, College Park University (Washington D. C.) in January 2005, I stopped this research for 2 years to dedicate time for mathematical problems with algebra for a generation of aggregates in temporal-probabilistic (TP) databases. We published two papers [13, 14], and I finished this work by a fundamental revision of the erroneous Subrahmaninan's fixed-point semantics of TP-logic programs [15].

So, only after 2007 I finished the development of categorial methods applied to data integration. This work was provided in my book [5] in 2014, based on the behavioral point of view for databases.[4] This book [5] presented partially the first time the *categorial symmetry* to wide scientific community, as a fundamental part of my PhD thesis, and hence a number of different people asked me the electronic copy of my original PhD thesis. But I did not have it after more than 16 years, and hence I could not satisfy their interest to this pure theoretical categorial work.

Thus, I finally decided to write this book by extending the original work developed in my PhD thesis in 1998. A lot of work in category theory in last 20 years offered me also the possibility to better explain the ideas and motivation, determined intuitively by my reflections more than 20 years ago. From this distance of time, these initial intuitions appear in a new and more clear light; so that this book is not a simple rewriting of my previous work, but a deeper insight on the fundamental nature of category theory and its hidden relationships with the human mind structure and functionalities developed during a long human evolution.

Roma 2022 Zoran Majkić

4 Seven years later, BookAuthority rated it as the best database theory book of all time, the best computer science, best big data and best database schema book as well.

Notation conventions

The symbol $\triangleq$ will be used for the explicit definition of the mathematical concepts, while the symbol "=" (also it will be used sometimes for the same fact when it is clear from the context) for the usual meaning of the equivalence of the left- and right-hand sides in the equations.

The symbol $\equiv$ denotes standard logical equivalence, while $\doteq$ for the binary identity predicate in the first-order logic. We denote by $R_{=}$ the Tarski's interpretation of $\doteq$. So, for the equality we will use the standard symbol $=$, while for different equivalence relations we will employ the symbols $\simeq, \approx, \eqsim$, etc. The term "iff" is used for "if and only if."

We use logical symbols both in our formal languages and in the metalanguage. The notation slightly differs: in classical propositional and FOL, we use $\wedge, \vee, \Rightarrow, \neg$ (and $\exists, \forall$ for the FOL quantifiers). As the metasymbol for conjunction, the symbol & will be used.

In our terminology, we distinguish functions (graphs of functions) and maps. A (graph of) *function* from X to Y is a binary relation $F \subseteq X \times Y$ (subset of Cartesian product of the sets X and Y) with domain A satisfying the functionality condition $(x,y) \in F \& (x,z) \in F$ implies $y = z$, and the triple $\langle f, X, Y\rangle$ is then called a *map* (or morphism in a category) from A to B, denoted by $f : X \to Y$ as well. The composition of functions is denoted by $g \cdot f$, so that $(g \cdot f)(x) = g(f(x))$, while the composition of mappings (in a given category) by $g \circ f$. $\mathcal{N}$ denotes the set of natural numbers. Here is some other set-theoretic notation:

- The symbol $\mathbb{N}$ is used for the set of natural numbers, while $\emptyset$ denotes the empty set;
- $\mathcal{P}(X)$ denotes the power set of a set X, and $X^n \triangleq \overbrace{X \times \cdots \times X}^{n}$ the n-ary Cartesian product, and Y^X denotes the set of all functions from X to Y;
- We use $\subseteq$ for inclusion, $\subset$ for proper inclusion (we use $\leq, \preceq, \sqsubseteq$ for partial orders), and $X \subseteq_\omega Y$ denotes that X is a *finite* subset of an infinite set Y. By $X \backslash Y$ or $X - Y$, we denote the set difference between the set X and set Y;
- R^{-1} is the converse of a binary relation $R \subseteq X \times Y$ and $\overline{R}$ is the complement of R (equal to $(X \times Y) \backslash R$);
- id_X is the identity map on a set X. $|X|$ denotes the cardinality of a set (or list) X;
- For a set of elements $x_1, \ldots, x_n \in X$, we denote by $\mathbf{x}$ the sequence (or *tuple*) $(x_1, \ldots, x_n)$, and if $n = 1$ simply by x_1, while for $n = 0$ the empty tuple $<>$. An n-ary relation R, with $n = \mathrm{ar}(R) \geq 1$, is a set (also empty) of tuples $\mathbf{x}_i$ with $|\mathbf{x}_i| = n$, with $<> \in R$ (the empty tuple is a tuple of every relation);
- By $\pi_{\mathbf{K}}(R)$, where $\mathbf{K} = [i_1, \ldots, i_n]$ is a sequence of indexes with $n = |\mathbf{K}| \geq 1$ we denote the projection of R with columns defined by ordering in $\mathbf{K}$. If $|\mathbf{K}| = 1$, we write simply $\pi_i(R)$.

A *relational symbol* (a predicate letter in FOL) r and its extension (relation table) R will be called often shortly as a "relation," where it is clear from the context. If R is the extension of a relational symbol r, we write $R = \|r\|$.

https://doi.org/10.1515/9783111081670-205

Contents

Preface —— VII

Dependencies between the chapters —— XVII

Detailed plan —— XIX

Acknowledgments and short history of this book —— XXV

Notation conventions —— XXVII

1 Basic transformations of categories: Hierarchy of *n*-dimensional levels —— 1
1.1 The *n*-dimensional levels: A nonset-based definition of natural numbers —— **1**
1.2 Comma lifting for *n*-dimensional levels —— **6**
1.3 Introduction to comma-induction in the *n*-dimensional hierarchy —— **16**
1.4 Comma-induction of the adjunctions —— **20**

2 Comma-propagation transformations: Global categorial symmetries —— 29
2.1 Introduction to general comma-propagation —— **29**
2.2 Comma-propagation of functors and natural transformations —— **34**
2.3 Comma-propagation of (co)limits —— **40**
2.4 Example: Comma-propagation and infinite hierarchy of small-complete categories —— **50**
2.5 An analogy between physical and abstract categorial Global symmetries: Adjunctions-as-fields —— **57**
2.5.1 Analogy between adjunctions and metric tensor field with Einstein–Hilbert action —— **60**
2.5.2 Comma-propagation transformation symmetries —— **65**

3 Arrows-to-objects conceptual transformation: Internal categorial symmetry —— 71
3.1 Introduction to categorial internal symmetry of primitive categorial concepts —— **71**
3.2 Internal symmetry and metacategory —— **77**
3.3 Conceptually closed categories: A topology —— **83**
3.4 Symmetry-extended categories —— **94**
3.5 Symmetry hierarchy upper bound: Imploded categories —— **101**

4 Internal symmetry and logical deduction —— 111
4.1 Natural deduction system —— **111**
4.1.1 First solution for tagging techniques of natural deduction —— **111**

4.1.2 Sequent-based solution for tagging techniques — **116**
4.2 Definition of symmetry-extended category **ND** for natural deduction — **119**
4.2.1 The properties of the covariant implication functor — **125**
4.2.2 Local Cartesian Closed Adjunction (pCCC) — **128**
4.2.3 The *n*-dimensional levels of natural deduction categories — **133**
4.3 Symmetry-extended category **IC** for propositional intuitionistic calculus — **138**

5 Internal symmetry and lambda calculus — 144
5.1 Introduction to lambda calculus — **144**
5.2 Reflexive objects in the Cartesian closed categories — **149**
5.3 Conceptually-closed CCC and its subcategories of idempotents — **156**
5.4 Internal categorial symmetry and fixed-point operators — **166**
5.5 Topological K-theory, idempotent completion and internal categorial symmetry — **169**

6 Internal symmetry and theory of processes: Strong bisimulation of computation trees — 173
6.1 Introduction to transition systems and their bisimulations — **173**
6.2 Regular languages, automata and internal categorial symmetry — **177**
6.3 Symmetry-extended category of finite labeled trees — **183**
6.4 Internal symmetry of the category of relations — **191**
6.5 Transition systems as fixed points in the *n*-dimensional level $\mathbf{Rel}_3$ — **196**
6.6 Strong bisimulations as fixed points in the *n*-dimensional level $\mathbf{Rel}_3$ — **201**
6.6.1 Mathematics via symmetry: Reduction of many valued into 2-valued logic — **207**
6.6.2 Many-valued knowledge invariance through modal logic transformations: Semantic reflection — **212**

7 Internal symmetry and data integration theory — 220
7.1 **DB** (Database) category — **220**
7.1.1 Morphism properties of **DB** category — **235**
7.1.2 Power-view endofunctor and monad T — **249**
7.1.3 Duality — **254**
7.2 Objects of **DB**: Basic operations and equivalence relations — **257**
7.2.1 Data federation operator in **DB** — **257**
7.2.2 Data separation operator in **DB** — **258**
7.2.3 The (strong) behavioral equivalence for databases — **261**
7.2.4 Weak observational equivalence for databases — **263**
7.3 Internal categorial symmetry of **DB** category — **266**
7.3.1 (Co)products — **270**
7.3.2 (Co)Limits and exponentiation — **274**

7.3.3 Kleisli semantics for database mappings — **282**
7.4 Partial ordering for databases — **288**
7.4.1 Matching tensor product — **292**
7.4.2 Merging operator — **295**
7.4.3 Universal algebra considerations — **298**
7.4.4 Algebraic database lattice — **302**
7.5 Enrichment — **313**
7.5.1 **DB** is a V-category enriched over itself — **315**
7.5.2 Internalized Yoneda embedding — **320**

A Appendix — 323
A.1 Introduction to lattices, algebras and logics — **323**
A.1.1 Introduction to deductive logic and binary sequent calculus — **327**
A.1.2 Introduction to first-order logic and Tarski's interpretations — **329**
A.1.3 Introduction to multimodal logics and Kripke semantics — **333**
A.2 Basic category theory — **334**
A.3 Introduction to RDB, database mappings and **DB** category — **348**
A.3.1 Basic database concepts — **351**
A.3.2 Database observations: Idempotent power-view operator — **357**
A.3.3 Logic versus algebras: Categorification by operads — **360**
A.3.4 Sketches and functors into the **DB** category — **363**
A.3.5 Semantics of DB schema mappings: Information fluxes — **370**
A.4 Introduction to field theory and symmetries — **381**
A.4.1 Vector fields on curved differentiable manifolds — **388**
A.4.2 Transformation of coordinates — **392**

Bibliography — 395

Index — 403

1 Basic transformations of categories: Hierarchy of *n*-dimensional levels

1.1 The *n*-dimensional levels: A nonset-based definition of natural numbers

The arrow categories are more simple forms of the *comma* categories and were introduced by Lawvere [16] in the context of the interdefinability of the universal concepts of category theory. The basic idea is the elevation of arrows of one category $\mathbf{C}$ to objects in another.

It is well known that, given a base category $\mathbf{C}$, we can represent its morphisms as objects by using a derived *arrow* category $\mathbf{C} \downarrow \mathbf{C}$ (a special case of the comma category), such that for a given arrow $f : a \to b$ between two objects $a, b \in \mathrm{Ob}_{\mathbf{C}}$, we have the object denoted by $\langle a, b, f\rangle \in \mathrm{Ob}_{\mathbf{C}\downarrow\mathbf{C}}$ in this arrow category. Moreover, from the fact that if the base category is just an arrow category $\mathbf{C} = \mathbf{D} \downarrow \mathbf{D}$, necessarily we have to consider the arrow category $(\mathbf{D} \downarrow \mathbf{D}) \downarrow (\mathbf{D} \downarrow \mathbf{D})$, etc. We need to consider all higher levels of the arrow categories of a given initial category. So, obvious topics in this theory are:

– The notion of hierarchy of arrow categories for a given base category $\mathbf{C}$ in (1), inductively defined for $n \geq 1$ by

$$(\mathbf{C} \downarrow \mathbf{C})^1 \triangleq \mathbf{C} \downarrow \mathbf{C}, \quad (\mathbf{C} \downarrow \mathbf{C})^{n+1} \triangleq (\mathbf{C} \downarrow \mathbf{C})^n \downarrow (\mathbf{C} \downarrow \mathbf{C})^n \tag{1.1}$$

  and the derived *n-dimensional levels* $\mathbf{C}_n = (\mathbf{C} \downarrow \mathbf{C})^{n-1}$, for $n \geq 2$, defined by (2), which arrows are composed by 2^{n-1} arrows of the 1-dimensional level $\mathbf{C}_1 = \mathbf{C}$.
– Questions concerning the existence of certain normal forms (comma lifting). They constitute the nucleus of this general *n*-dimensional level theory, as it will be explained in next.

The concept of "comma lifting" is fundamental for the process of comma-propagation (and for its particular case of the comma-induction), which determines how the covariant functors, between basic categories in question, and the relative natural transformations, are inherited between all *n*-dimensional levels of these basic categories. In what follows, will be provided the demonstrations of the important properties of the comma-propagation, as the closure property and the general conservation of adjunctions, limits and colimits, within the *n*-dimensional levels.

In what follows, we will use standard comma-projection functors for all *n*-dimensional levels of a given base category $\mathbf{C}$, and $n \geq 1$:

– The first comma projection, $F_{\mathrm{st}} = (F^0_{\mathrm{st}}, F^1_{\mathrm{st}}) : \mathbf{C}_{n+1} \to \mathbf{C}_n$, such that:
  for each object $\langle a, b, f\rangle$ in $\mathbf{C}_{n+1}$, $F_{\mathrm{st}}(\langle a, b, f\rangle) = F^0_{\mathrm{st}}(\langle a, b, f\rangle) = a$;
  for each arrow (h_1, h_2) in $\mathbf{C}_{n+1}$, $F_{\mathrm{st}}(h_1, h_2) = F^1_{\mathrm{st}}(h_1, h_2) = h_1$.
– The second projection, $S_{\mathrm{nd}} = (S^0_{\mathrm{nd}}, S^1_{\mathrm{nd}}) : \mathbf{C}_{n+1} \to \mathbf{C}_n$, such that:

https://doi.org/10.1515/9783111081670-001

for each object $\langle a,b,f\rangle$ in $\mathbf{C}_{n+1}$, $S_{\text{nd}}(\langle a,b,f\rangle) = S^0_{\text{nd}}(\langle a,b,f\rangle) = b$;
for each arrow (h_1,h_2) in $\mathbf{C}_{n+1}$, $S_{\text{nd}}(h_1,h_2) = S^1_{\text{nd}}(h_1,h_2) = h_2$.

– The natural transformation (introduced by commutative diagram (A.8) in Section A.2), $\psi : F_{\text{st}} \xrightarrow{\cdot} S_{\text{nd}}$.
Note that a natural transformation ψ associate to every object X an arrow $\psi_X : F^0_{\text{st}}(X) \to S^0_{\text{nd}}(X)$, and this mapping can be represented by the function denoted by

$$J^{-1} : \text{Ob}_{\mathbf{C}_{n+1}} \to \text{Mor}_{\mathbf{C}_n}. \tag{1.2}$$

The example of this natural transformation ψ is provided in scheme (3.1) where the arrow components of this natural transformation are the arrows $J^{-1}(\langle a,b,f\rangle) = \psi_{\langle a,b,f\rangle} = f$ and $J^{-1}(\langle c,d,g\rangle) = \psi_{\langle d,d,g\rangle} = g$, represented in the commutative diagram in (1.6).

Thus, as emphasized previously, the general category theory is oriented dominantly to the morphisms (arrows) between the objects than to objects (the properties of the objects are expressed by the arrow for which it is domain or codomain), and hence categorial language has this general transformation of the objects into the arrows (by using the natural transformation between functors). We are able to inverse this process and to derive the objects from the natural transformation arrows, in a general way.

Definition 1. Let us extend the (general) concept of natural transformation $\eta : F \xrightarrow{\cdot} G$ in diagram A.8, Section A.2, between two functors $F, G : \mathbf{B} \to \mathbf{D}$,

$$\begin{array}{ccc} F(X) & \xrightarrow{\eta_X} & G(X) \\ {\scriptstyle F(m)}\downarrow & & \downarrow{\scriptstyle G(m)} \\ F(Y) & \xrightarrow{\eta_Y} & G(Y) \end{array} \quad \Leftarrow F,G \quad \begin{array}{c} X \\ \downarrow{\scriptstyle m} \\ Y \end{array} \quad \begin{array}{ccc} \langle F(X), G(X), \eta_X\rangle & & \\ H \mapsto & \downarrow{\scriptstyle (F(m);G(m))} & \\ \langle F(Y), G(Y), \eta_Y\rangle & & \end{array} \tag{1.3}$$

in $\mathbf{D}$ in $\mathbf{B}$ in $(F \downarrow G)$

by the introduction of a new functor $H = (H^0, H^1) : \mathbf{B} \to (F \downarrow G)$ into the comma category $(F \downarrow G)$, where $H^0 = (F^0, G^0, \eta)$ is the triple of functions (where η is considered as the function from objects in $\mathbf{B}$ into morphisms in $\mathbf{D}$) such that $H^0(X) = \langle F(X), G(X), \eta_X\rangle$ and the couple of functions $H^1 = (F^1; G^1)$ such that $H^1(m) = (F(m); G(m))$, as shown in the diagram above.

From the fact that $F(X) = \text{dom}(\eta_X)$ and $G(X) = \text{cod}(\eta_X)$, we have that the object $H^0(X) = \langle F(X), G(X), \eta_X\rangle = J(\eta_X)$ by defining this operator as the triple of functions for arrows

$$J \triangleq \langle \text{dom}, \text{cod}, \text{id}\rangle : \text{Mor}_\eta \to \text{Ob}_{F\downarrow G} \tag{1.4}$$

where Mor_η is the subset of arrows in $\mathbf{D}$, which are components of the natural transformation η.

Thus, now when we investigate the full duality (internal symmetry) between the arrows and objects of categories, we need the introduction of a new basic operator, which transforms the arrows of a given category into the objects of the corresponding arrow (comma) category. Moreover, we also introduce this set of new operators, which will be used for the n-dimensional levels of a given base category $\mathbf{C}$.

Definition 2. Let $\mathbf{C}$ be a base category with n-dimensional levels $\mathbf{C}_n$, $n \geq 1$ with functor-projections $F_{\mathrm{st}}, S_{\mathrm{nd}} : \mathbf{C}_{n+1} \to \mathbf{C}_n$ and the following operators:

1. Operator J^{-1} in (1.2) corresponding to the fundamental natural transformation

$$\psi : F_{\mathrm{st}} \xrightarrow{\cdot} S_{\mathrm{nd}}. \tag{1.5}$$

2. Encapsulation operator J in (1.4):
so from Definition 1, in the case when $\mathbf{B} = \mathbf{C}_{n+1} = \mathbf{C}_n \downarrow \mathbf{C}_n$, $\mathbf{D} = \mathbf{C}_n$, functors $F = F_{\mathrm{st}}$ and $G = S_{\mathrm{nd}}$ so that $\eta = \psi$ and H are just the identity endofunctor of $\mathbf{C}_{n+1}$, we obtain the function $J : \mathrm{Mor}_{\mathbf{C}_n} \to \mathrm{Ob}_{\mathbf{C}_{n+1}}$, inverse to the function J^{-1} in (1.2), such that for each arrow $f : a \to b$ in $\mathbf{C}_n$, we obtain $J(f) = \langle a, b, f \rangle$.

Let us verify that the encapsulation operator J in the definition above is well-defined, by considering that the general diagram (1.3) reduces to the following specific diagram:

$$\begin{array}{ccc} a & \xrightarrow{\eta_X = f} & b \\ {\scriptstyle h}\downarrow & & \downarrow{\scriptstyle k} \\ c & \xrightarrow{\eta_Y = g} & d \end{array} \qquad \begin{array}{c} X = \langle a, b, f \rangle \\ \Leftarrow F_{\mathrm{st}}, S_{\mathrm{nd}} \;\downarrow (h;k) \\ Y = \langle c, d, g \rangle \end{array} \qquad \begin{array}{c} H^0(X) = X = J(f) \\ H = \mathrm{Id} \mapsto \downarrow (h;k) \\ H^0(Y) = Y = J(g) \end{array} \tag{1.6}$$

$$\text{in } \mathbf{D} = \mathbf{C}_n \qquad \text{in } \mathbf{B} = \mathbf{C}_{n+1} \qquad \text{in } (F \downarrow G) = \mathbf{C}_{n+1}$$

so that the arrow $f : a \to b$ in $\mathbf{D} = \mathbf{C}_n$ is mapped into the object $J(f)$ in the category $\mathbf{C}_{n+1}$, with $J(f) = H^0(X) = H^0(\langle a, b, f \rangle) = \langle a, b, f \rangle$.

Let us show that the set of arrow components of the natural transformation $\eta = \psi : F_{\mathrm{st}} \xrightarrow{\cdot} S_{\mathrm{nd}}$ is just the set of all arrows in $\mathbf{D} = \mathbf{C}_n$, just because for each arrow $f : a \to b$ in $\mathrm{Mor}_{\mathbf{C}_n}$ in the category $\mathbf{B} = \mathbf{C}_{n+1}$ we have always the identity arrow $(\mathrm{id}_a; \mathrm{id}_b) : X \to X$ of the object $X = \langle a, b, f \rangle$, and hence the arrow $f = \psi_X$ is the component of the natural transformation $\eta = \psi$, i. e., $\mathrm{Mor}_\eta = \mathrm{Mor}_\psi = \mathrm{Mor}_{\mathbf{C}_n}$, so that from (1.4) and the fact that comma category $F \downarrow G$ (for $F = F_{\mathrm{st}}$ and $G = S_{\mathrm{nd}}$) is just equal to category $\mathbf{B} = \mathbf{C}_{n+1}$, the function $J : \mathrm{Mor}_{\mathbf{C}_n} \to \mathrm{Ob}_{\mathbf{C}_{n+1}}$ has correct domain and codomain.

So, from the diagram above, we have that $J^{-1}J(f) = f$, i. e., $J^{-1}J$ is the identity. Analogously, $JJ^{-1}(\langle a, b, f\rangle) = J(f) = \langle a, b, f\rangle$, so that also JJ^{-1} is an identity and hence the operator J of encapsulation of arrows into objects and J^{-1} of deencapsulation are mutually inverse operators.

More about categorial definition of natural numbers

We have introduced in preface the new categorial definition of natural numbers by the inductively defined pattern of *n*-dimensional levels of the minimal preorder category **1** (with single object denoted by • and only identity arrow id of it). Let us consider in more details this nonset based definition of the set of natural numbers.

To render more easy this presentation, in the place of single identity arrow $\mathrm{id} : \bullet \to \bullet$ the labeled arrows like, $f_i, g_i, h_i, k_i, \ldots$. Having this in mind, that each labeled arrow is equal to the identity arrow, we obtain the following multidimensional representation of natural numbers:

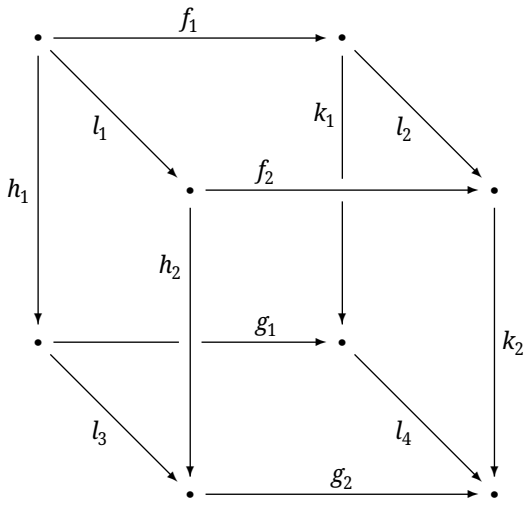
1. The number zero equal to empty category **0** geometrically is zero-dimensional (without any dimension);
2. The number 1, equal to the category **1**, where each (i. e., unique) arrow is *one-dimensional* (geometrically a linear line segment)

$$\bullet \xrightarrow{\mathrm{id}} \bullet$$

3. The number 2, equal to the arrow category $\mathbf{1} \downarrow \mathbf{1}$, where each (again unique) arrow $(h_1; k_1) : J(f_1) \to J(g_1)$ where $J(f_1) = J(g_1)$ are equal to $J(\mathrm{id})$ (the unique object in $\mathbf{1} \downarrow \mathbf{1}$), which in base category **1** must satisfy the equation $k_1 \circ f_1 = g_1 \circ h_1$, that is a commutative diagram geometrically represented in a *bidimensional* surface

$$\begin{array}{ccc} \bullet & \xrightarrow{f_1} & \bullet \\ {\scriptstyle h_1}\downarrow & & \downarrow{\scriptstyle k_1} \\ \bullet & \xrightarrow{f_1} & \bullet \end{array}$$

4. The number 3, equal to the arrow category $(\mathbf{1} \downarrow \mathbf{1}) \downarrow (\mathbf{1} \downarrow \mathbf{1})$, where each (again unique) arrow $((l_1; l_2); (l_3; l_4)) : J(h_1; k_1) \to J(h_2; k_2)$ where $J(h_1; k_1) = J(h_2; k_2)$ are equal to $J(\mathrm{id}; \mathrm{id})$ (the unique object in $(\mathbf{1} \downarrow \mathbf{1}) \downarrow (\mathbf{1} \downarrow \mathbf{1})$), which in base category **1** must satisfy the six equations (of each external surface of the cube in next diagram), that is commutative diagrams geometrically represented in the following *3-dimensional* cube:



representing six *equations*, the first two represented by the objects $J(h_1; k_1)$ and $J(h_2; k_2)$ and the next four derived from the arrow $((l_1; l_2); (l_3; l_4))$ from the first into second object:

$$g_1 \circ h_1 = k_1 \circ f_1$$
$$g_2 \circ h_2 = k_2 \circ f_2$$
$$f_2 \circ l_1 = l_2 \circ f_1$$
$$h_2 \circ l_1 = l_3 \circ h_1$$
$$k_2 \circ l_2 = l_4 \circ k_1$$
$$g_2 \circ l_3 = l_4 \circ g_1$$

5. etc.

Note that this definition is multidimensional, and not set-based, because the commutative diagrams are not the sets but the complex composition of arrows. More over, differently from the von Neumann set-based definition, we do not introduce the "member of" relation, and in fact $0 \in 1$, and $1 \in 2$ does not hold for this categorial multidimensional commutative-diagram's definition of natural numbers.

The commutative diagrams above distinguish clearly the category theory from the set-based theory but also from the pure geometric theory: it is true that geometrically these commutative diagrams have *n-dimensional forms*, but they are also commutative, and hence represents the *algebraic equations*. I remarked previously that this pattern for definition of natural numbers can be defined for *each* finite or infinite category **C**, so that in that case we have not only the commutative diagrams in such categories composed by only identity arrows (the trivial case), but also by the nonidentity arrows. That is, the algebraic equations of the commutative diagrams in the pattern above are not trivial

only but can be significant for a model theory represented by such a category **C** different from the minimal nonempty category **1**.

From my point of view, this pattern above demonstrates clearly how the category theory as a particular synthesis of geometric and algebraic equational theory, is distinguished (its own) mathematical theory different both from the pure geometry and from the set-based mathematics.

We have provided not only a nonset-theoretical definition of natural numbers, but have shown how the commutative diagrams in any base category **C** are the sources for a generation of the multidimensional cubes, and this basic categorial "spatial" (geometric) topology inside this base category is classified by its *n*-dimensional arrow categories: in each *n*-dimensional arrow category derived from a base category **C**, we have only the arrows that represent the *n*-dimensional commutative cubes of the base category **C**. So, topologically a *n*-dimensional level represents a subcategory of **C**, composed by the *n*-dimensional commutative cubes.

This categorial partition of a base category is categorial analogous to the partition of a given set S by its powerset $\mathcal{P}(S)$ whose elements are all subsets of S. So in this set-based framework, we can define the *n*-dimensional levels S_n inductively by $S_0 = \emptyset, S_1 = S$ and $S_{n+1} = S_n \times S$ for $n \geq 1$, where "×" is Cartesian product for sets. So we can define by set-based pattern the natural numbers (different from von Neumann's definition, but analog to our categorial method provided previously), by using minimal nonempty (singleton) set $S = \{*\}$ composed by the single element '$*$',

$$0 = \emptyset$$
$$1 = S$$
$$2 = S \times S$$
$$3 = S \times S \times S$$

etc. That is, each set-based *n*-dimensional level S_n is composed by *n*-ary elements of the base set S. It is easy now to verify fundamental difference of such a set-based pattern from the categorial *n*-dimensional pattern of definition of natural numbers.

1.2 Comma lifting for *n*-dimensional levels

The main concept in the *n*-dimensional levels theory of arrow categories is the fundamental coma-propagation process, based on the coherent translation of the categories, functors and natural transformations for a given *n*-dimensional into next $(n+1)$-dimensional level of this stratified theory.

We will use the incrementing symbol '$\hat{_}$' to denote such "coma lifting" of the categories, functors between them and natural transformations between such functors, as follows:

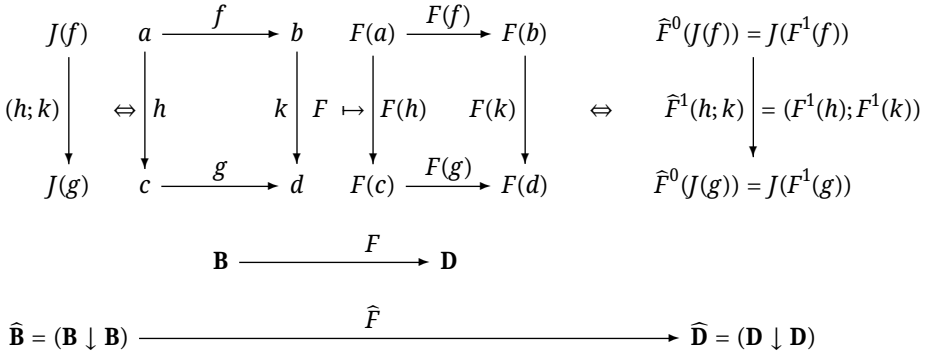
- For a category **D**, its comma lifting is $\widehat{\mathbf{D}} \triangleq \mathbf{D} \downarrow \mathbf{D}$;
- For a covariant functor $F : \mathbf{B} \to \mathbf{D}$, its comma-lifting functor is $\widehat{F} : \widehat{\mathbf{B}} \to \widehat{\mathbf{D}}$;
- For a natural transformation $\tau : F \xrightarrow{\cdot} G$, its comma-lifting natural transformation is $\widehat{\tau} : \widehat{F} \xrightarrow{\cdot} \widehat{G}$.

Note that the comma lifting represents the transfer of the semantics from a given theory into its metatheory. It is well known that the functors $F : \mathbf{B} \to \mathbf{D}$ can be used as mathematical structures that can express how the mathematical structures (usually expressed by diagrams of a category **B**) are mapped (by this functor) into the denotational-semantics category **D** (as the **Set** category, for example), by defining the semantics of the given structures of this theory defined in the category **B**. Then the comma lifting shows how this semantics of a given theory in **B** propagates into its metatheory in the comma lifted (higher-dimensional level) category $\widehat{\mathbf{B}} = (\mathbf{B} \downarrow \mathbf{B})$.

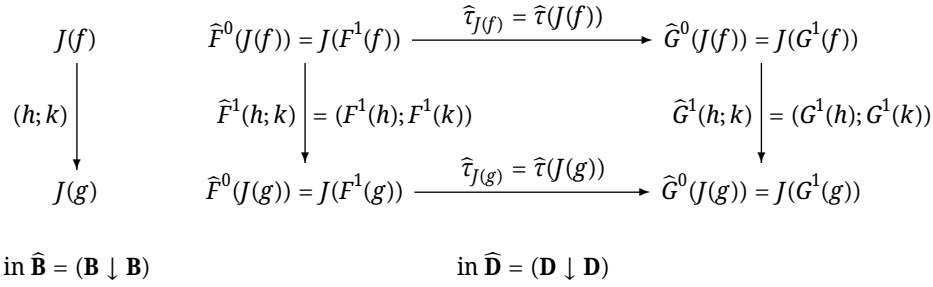
This fact is easy to understand if we consider that each arrow $f : a \to b$ in the "theory" category **B**, equivalent to the commutative diagram $f \circ \mathrm{id}_a = \mathrm{id}_b \circ f$ where id_a and id_b are the identity arrows of the domain and codomain objects of this arrow f, corresponds exactly to the arrow $(f;f) : J(\mathrm{id}_a) \to J(\mathrm{id}_b)$ in the comma-lifted metatheory category $\widehat{\mathbf{B}} = (\mathbf{B} \downarrow \mathbf{B})$, and by using this correspondence, for each commutative diagram in the theory category **B** (expressing some property of this theory) there exists exactly the same commutative diagram in the metatheory (comma-lifted) category $\widehat{\mathbf{B}} = (\mathbf{B} \downarrow \mathbf{B})$, by substituting each arrow f in this diagram in theory category **B**, by the arrow $(f;f)$ in the metatheory category $\widehat{\mathbf{B}} = (\mathbf{B} \downarrow \mathbf{B})$. Consequently, by this process of comma lifting we support the propagation of the properties of a given theory to all higher metatheories of this base theory.

Of course, the metatheory (comma-lifted) category is more expressive than base theory category, because its objects are generally the arrows in the theory category, and hence an arrow in the metatheory category can represent *any* (not only the banal, by using the identity arrows, as explained previously) *commutative diagram* (i. e., some *equation*, like $k \circ f = g \circ h$ in the scheme bellow). Arrows of the higher metatheories express the *sets of intercorrelated equations* expressed in base theory category: for example, the arrows in the metametatheory category $(\mathbf{B} \downarrow \mathbf{B}) \downarrow (\mathbf{B} \downarrow \mathbf{B})$ represent the set of four equations (in theory **B**) that compose a commutative cube with six correlated equations (each equation corresponds to one 2-D side of such a cube where vertex are the objects of the base theory category; such cubes of correlated equations there exist always by using also the identity arrows). These facts explain why the *n*-levels category theory has a general importance in the mathematics.

The following scheme show the comma lifting of the functor $F : \mathbf{B} \to \mathbf{D}$ (between two commutative diagrams in the center of this scheme) into the functor $\widehat{F} : \widehat{\mathbf{B}} \to \widehat{\mathbf{D}}$ (from left hand side arrow into right-hand side arrow):



The comma-lifting $\widehat{\tau}$ of the natural transformation $\tau : F \dot{\longrightarrow} G$ can be represented by the following scheme (in next we will use the representation of natural transformations as *functions* from the object of the domain category into the arrows of the codomain category of functors, i. e., instead of its components τ_a for an object a, by $\tau(a)$):



where the components of this comma-lifted natural transformation are $\widehat{\tau}_{J(f)} = \widehat{\tau}(J(f)) \triangleq (\tau F^0_{\mathrm{st}}; \tau S^0_{\mathrm{nd}})(J(f))$, i. e., in functional representation of natural transformations, we have that the comma-lifted transformation is the following couple of functions:

$$\widehat{\tau} \triangleq (\tau F^0_{\mathrm{st}}; \tau S^0_{\mathrm{nd}}) : \mathrm{Ob}_{\mathbf{B}\downarrow\mathbf{B}} \to \mathrm{Mor}_{\mathbf{D}\downarrow\mathbf{D}}.$$

Example 1. Note that the commutative diagram in the scheme above is just a commutative cube in the category **D** composed by the following six equations:

1. Functorial consequences of the same base "theory" equation $k \circ f = g \circ h$: the $F(k) \circ F(f) = F(g) \circ F(h)$ and $G(k) \circ G(f) = G(g) \circ G(h)$ on the two opposite sides of the cube;
2. Consequences of the two arrows' components of the "theory" natural transformation τ for the arrows $f : a \to b$ and $g : c \to d$ in **B**: the $G(f) \circ \tau(F^0_{\mathrm{st}} J(f)) = G(f) \circ \tau(a) = \tau(b) \circ F(f) = \tau(S^0_{\mathrm{nd}} J(f)) \circ F(f)$ obtained from the arrow $f : a \to b$, $G(g) \circ \tau(F^0_{\mathrm{st}} J(g)) = G(g) \circ \tau(c) = \tau(d) \circ F(g) = \tau(S^0_{\mathrm{nd}} J(g)) \circ F(g)$, on the two opposite sides of the cube;
3. Consequences of the two arrows' components of the "theory" natural transformation τ for the arrows $h : a \to c$ and $k : b \to d$ in **B**: the $G(h) \circ \tau(F^0_{\mathrm{st}} J(f)) = G(h) \circ \tau(a) =$

$\tau(c) \circ F(h) = \tau(F^0_{\mathrm{st}} J(g)) \circ F(h)$ and $G(k) \circ \tau(S^0_{\mathrm{nd}} J(f)) = G(k) \circ \tau(b) = \tau(d) \circ F(k) = \tau(S^0_{\mathrm{nd}} J(g)) \circ F(k)$, on the two opposite sides of the cube.

Remark. Thus, all equations are derived by natural transformation and two functors, from the unique base "theory" equation (commutative diagram) $k \circ f = g \circ h$ in **B**, so that the "set of correlated equations" is this result of the action of the natural transformation τ and its two functors F and G.

So, as shown in the last scheme above, from the fact that the (unique) arrow $(h; k) : J(f) \to J(g)$ in the comma-lifted category (metatheory) $\widehat{\mathbf{B}} = (\mathbf{B} \downarrow \mathbf{B})$ is just an equation (commutative diagram in "theory" **B**), the commutative diagram obtained from this arrow by action of the comma-lifted natural transformation $\widehat{\tau}$ in effect represents these six correlated equations (commutative diagrams in the category **D**) in a compact representation by a unique commutative diagram in the comma-lifted $\widehat{\mathbf{D}} = (\mathbf{D} \downarrow \mathbf{D})$. □

Let us show that this process of comma lifting for functors and natural transformations is well-defined.

Proposition 1. *The operator of "comma lifting" is defined for the categories by* $\widehat{\mathbf{B}} = (\mathbf{B} \downarrow \mathbf{B})$. *There exist the following functors and natural transformation as a comma-lifting consequence in the higher n-dimensional levels:*

1. *For each covariant functor* $F = (F^1, F^2) : \mathbf{B} \to \mathbf{D}$, *there exists the following comma-lifted functor:*

$$\widehat{F} \triangleq (JF^1\psi, (F^1F^1_{\mathrm{st}}; F^1S^1_{\mathrm{nd}})) : \widehat{\mathbf{B}} \to \widehat{\mathbf{D}}, \tag{1.7}$$

and for each natural transformation $\tau : F \xrightarrow{\cdot} G$ *there exists the following comma-lifted natural transformation:*

$$\widehat{\tau} \triangleq (\tau F^0_{\mathrm{st}}; \tau S^0_{\mathrm{nd}}) : \widehat{F} \xrightarrow{\cdot} \widehat{G}, \tag{1.8}$$

and the following functor denominated "nat-functor"

$$N_\tau \triangleq (J\tau, \widehat{\psi} J\widehat{\tau} J) : \mathbf{B} \to (\mathbf{D} \downarrow \mathbf{D}) \tag{1.9}$$

where the natural transformation $\widehat{\psi} : \widehat{F}_{\mathrm{st}} \xrightarrow{\cdot} \widehat{S}_{\mathrm{nd}}$ *is the comma-lifted* fundamental *natural transformation in* (1.5).

2. *For a natural transformation* ψ *in* (1.5) *with* $F_{\mathrm{st}}, S_{\mathrm{nd}} : \mathbf{C}_{n+1} \to \mathbf{C}_n$ *with the comma-lifted natural transformation* $\widehat{\psi} : \widehat{F}_{\mathrm{st}} \xrightarrow{\cdot} \widehat{S}_{\mathrm{nd}}$, *we define the exchange object operator* $\Psi \triangleq J\widehat{\psi} : \mathrm{Ob}_{\mathbf{C}_{n+2}} \to \mathrm{Ob}_{\mathbf{C}_{n+2}}$ *and the arrow exchange operator* $\Phi \triangleq ((F^1_{\mathrm{st}}\widehat{F}^1_{\mathrm{st}}; F^1_{\mathrm{st}}\widehat{S}^1_{\mathrm{nd}}); (S^1_{\mathrm{nd}}\widehat{F}^1_{\mathrm{st}}; S^1_{\mathrm{nd}}\widehat{S}^1_{\mathrm{nd}})) : \mathrm{Mor}_{\mathbf{C}_{n+2}} \to \mathrm{Mor}_{\mathbf{C}_{n+2}}$. *They compose the "duality" endofunctor*

$$L_\perp \triangleq (\Psi, \Phi) : \mathbf{C}_{n+2} \to \mathbf{C}_{n+2} \tag{1.10}$$

such that the composition $L_\perp \circ L_\perp$ is the identity functor. This duality endofunctor is a particular case of the nat-functor (1.9), *when* $\mathbf{B} = \mathbf{C}_{n+2}$, $\mathbf{D} = \mathbf{C}_{n+1}$, $F = \widehat{F}_{\mathrm{st}}$ *and* $G = \widehat{S}_{\mathrm{nd}}$, *natural transformation* $\tau = \widehat{\psi}$, *so that* $L_\perp = N_\tau = (J\tau, \widehat{\psi}J\tau J)$.

Proof. It is enough to show the functorial property, for the composition of arrows, for the lifted functor. In fact, by considering the diagram of the comma-lifted functor $\widehat{F}$, we have that $\widehat{F}^1(h_1 \circ h; k_1 \circ k) = (F^1(h_1 \circ h); F^1(k_1 \circ k)) =$ (from the functorial property of basic functor F) $= (F^1(h_1) \circ F^1(h); F^1(k_1) \circ F^1(k)) = (F^1(h_1); F^1(k_1)) \circ (F^1(h); F^1(k)) = \widehat{F}^1(h_1; k_1) \circ \widehat{F}^1(h; k)$.

The comma-lifted natural transformation $\widehat{\tau}$ is well-defined: the diagram in that scheme $\widehat{G}^1(h; k) \circ \widehat{\tau}(J(f)) = \widehat{\tau}(J(g)) \circ \widehat{G}^1(h; k)$ (is composed by two commutative diagrams in point 3 of Example 1).

For the functor N_τ in (1.9), we have that for any arrow $f : a \to b$ in $\mathbf{B}$, we obtain that

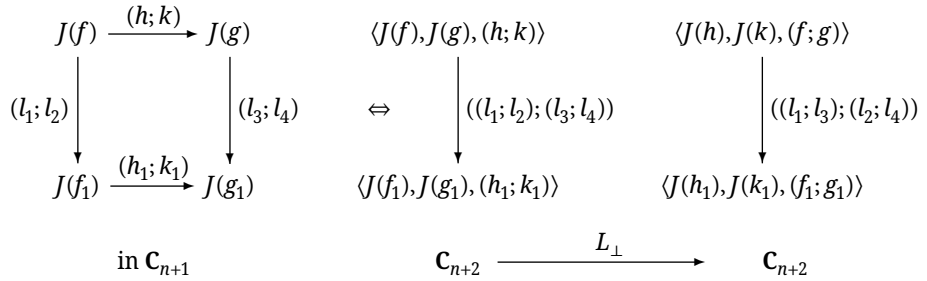
$$\begin{aligned} N^1_\tau(f) &= \widehat{\psi}J\widehat{\tau}J(f) = \widehat{\psi}J(\tau(a); \tau(b)) \\ &= (\psi F^0_{\mathrm{st}}; \psi S^0_{\mathrm{nd}})(J(\tau(a); \tau(b))) \\ &= (\psi(\langle F^0(a), G^0(a), F^1(f)\rangle); \psi(\langle F^0(b), G^0(b), G^1(f)\rangle)) \\ &= (F^1(f); G^1(f)). \end{aligned}$$

So, property of N_τ reduces to the functorial properties of the functors F and G, i. e., it satisfies the functorial property for the composition of arrows as well.

The arrow component of the functor $L_\perp$ is a 4-function based on the arrow components of the projection functors, so it satisfies the functorial property for the composition of arrows, i. e., $L_\perp$ is a well-defined functor, with:

1. The exchange object operator $\Psi \triangleq J\widehat{\psi}$ is the function $\Psi : \mathrm{Ob}_{\mathbf{C}_{n+2}} \to \mathrm{Ob}_{\mathbf{C}_{n+2}}$, which exchanges an object $\langle J(f), J(g), (h; k)\rangle$ in $(n+2)$-dimensional level $\mathbf{C}_{n+2}$ by its dual object $\langle J(h), J(k), (f; g)\rangle$, derived from a commutative 2-D diagram $k \circ f = g \circ h$ in $\mathbf{C}_{n+1}$. Hence, Ψ^2 is the identity function.

2. The exchange arrow operator $\Phi \triangleq ((F^1_{\mathrm{st}}F^1_{\mathrm{st}}; F^1_{\mathrm{st}}S^1_{\mathrm{nd}}); (S^1_{\mathrm{nd}}F^1_{\mathrm{st}}; S^1_{\mathrm{nd}}S^1_{\mathrm{nd}}))$ is a four-component function (divided by ';'), $\Phi : \mathrm{Mor}_{\mathbf{C}_{n+2}} \to \mathrm{Mor}_{\mathbf{C}_{n+2}}$, which exchanges four arrows all directed from a 2-D commutative diagram $k \circ f = g \circ h$ (represented as an object in the higher level) into another 2-D commutative diagram (represented as an object in the higher level), from a commutative 3-D (cube) diagram (a cube) of arrows in $\mathbf{C}_{n+1}$. Hence, Φ^2 is identity function as well, and from point 1 above, we obtain that the composition $L_\perp \circ L_\perp$ is the identity functor. The action of this duality endofunctor is evident from this scheme,



in the way that the first arrow $((l_1;l_2);(l_3;l_4))$ represents the four equations (commutative diagrams in $\mathbf{C}_n$) and its dual arrow $((l_1;l_3);(l_2;l_4)) = L_\perp((l_1;l_2);(l_3;l_4))$ represents four other equations with two of them nonpresent in the first arrow; so both arrows, original and its dual, represent the six equations of a commutative cube in $\mathbf{C}_n$. The two equations corresponding to two horizontal arrows of the commutative diagram (in $\mathbf{C}_{n+1}$) remain invariant by the action of this duality functor (they are base and top (opposite) side of the commutative cube), so that the action of $H_\perp$ makes only the rotation for 90^0 of this cube presenting two other opposite vertical sides (equations) of it. With another rotation of 90^0, we obtain two visible vertical sides of the cube, which was visible at the beginning, and because of that the composite functor $L_\perp \circ L_\perp$ is just the identity functor (duality of dual arrow is just the original arrow). □

Note that we denominate the functor N_τ by the "nat-functor" because it is a direct consequence of the natural transformation τ. We have seen that the duality functor is a particular case of such general nat-functor, so let us consider the other examples of this nat-functor.

Example 2. Let us consider the following cases of the general nat-functor N_τ in (1.9):

1. The case when $\mathbf{B} = \mathbf{D}$, the nat-functor N_{τ_I} in (1.9), is obtained from the identity natural transformation $\tau_I : \mathrm{Id}_{\mathbf{C}} \xrightarrow{\cdot} \mathrm{Id}_{\mathbf{C}}$ (obtained from the identity functor when $F = G = \mathrm{Id}_{\mathbf{C}} : \mathbf{C} \to \mathbf{C}$). It is easy to show that this nat-functor is equal to the well-known *diagonal* functor (we will only substitute a category $\mathbf{B}$ by $\mathbf{C}$, in order to obtain the same original syntax used for n-dimensional levels). So from (1.9),

$$\blacktriangle = (J\tau_I, \widehat{\psi}J\widehat{\tau}_I J) : \mathbf{C} \to (\mathbf{C} \downarrow \mathbf{C}) \tag{1.11}$$

such that for any object $a \in \mathbf{C}$, it holds that $\blacktriangle^0(a) = J\tau_I(a) = J(\mathrm{id}_a) = \langle a, a, \mathrm{id}_a\rangle$, and for each arrow $f : a \to b$, from the derivation in the prof above, we obtain

$$\blacktriangle^1(f) = \widehat{\psi}J\widehat{\tau}_I J(f) = (F^1;G^1)(f) = (\mathrm{Id}^1_{\mathbf{C}};\mathrm{Id}^1_{\mathbf{C}})(f) = (f;f).$$

2. The case when $\mathbf{B} = (\mathbf{D} \downarrow \mathbf{D})$, with functors $F = F_{\mathrm{st}} : (\mathbf{D} \downarrow \mathbf{D}) \to \mathbf{D}$, $G = S_{\mathrm{nd}} : (\mathbf{D} \downarrow \mathbf{D}) \to \mathbf{D}$ and $\tau = \psi$.
   In this case, we obtain that N_τ is the identity endofunctor of $(\mathbf{D} \downarrow \mathbf{D})$. In fact, for any arrow $(f;g) : \langle a, c, h\rangle \to \langle b, d, k\rangle$ in $(\mathbf{D} \downarrow \mathbf{D})$, we have that

$$N^0_\tau(\langle a,c,h\rangle) = J\psi(\langle a,c,h\rangle) = J(h) = \langle a,c,h\rangle,$$

while for the arrow we have that

$$\begin{aligned} N^1_\tau(f;g) &= \widehat{\psi}J\widehat{\tau}J(f;g) = \widehat{\psi}J\widehat{\psi}J(f;g) = \widehat{\psi}J(\psi F^0_{\mathrm{st}};\psi S\mathrm{nd}^0)J(f;g) \\ &= \widehat{\psi}J(h;k) = (\psi F^0_{\mathrm{st}};\psi S^0_{\mathrm{nd}})J(h;k) \\ &= (f;g). \end{aligned}$$

3. The case when $\mathbf{B} = \mathbf{D}$ is the category with the terminal object $\top$, so for F equal to the identity endofunctor, and constant endofunctor G (which maps each object $a \in \mathrm{Ob}_D$ into this terminal object $\top \in \mathrm{Ob}_D$ and each arrow $f : a \to b$ into the identity arrow $\mathrm{id}_\top : \top \to \top$), we obtain the natural transformation τ such that for each object a its component $\tau(a) : a \to \top$ is a unique arrow from this object a into the terminal object $\top$. So that, for each arrow $f : a \to b$ in $\mathbf{D}$, we obtain

$$\begin{aligned} N_\tau(f) &= \widehat{\psi}J\widehat{\tau}J(f) = \widehat{\psi}J(\tau(a);\tau(b)) = \widehat{\psi}(\langle J(f), J(\mathrm{id}_\top), (\tau(a);\tau(b))\rangle) \\ &= (\psi F^0_{\mathrm{st}};\psi S^0_{\mathrm{nd}}(\langle J(f), J(\mathrm{id}_\top), (\tau(a);\tau(b))\rangle) = (\psi J(f);\psi J(\mathrm{id}_\top)) \\ &= (f;\mathrm{id}_\top) \in \mathrm{Mor}_{\mathbf{D}\downarrow\mathbf{D}}, \end{aligned}$$

which is an equivalent representation of the arrow f in $\mathbf{D}$ in the arrow category $\mathbf{D} \downarrow \mathbf{D}$.

We have the following property for the comma lifting of categories and functors between them.

Corollary 1. *The operator of "comma lifting" defined in Proposition* 1 *satisfies the following property: for each functor F, which is an arrow* $F : \mathbf{B} \to \mathbf{D}$ *in* **Cat** *from a category* **B** *into a category* **D**, *the following diagram in* **Cat** *commutes:*

$$\begin{array}{ccc} \mathbf{B} & \xrightarrow{\blacktriangle} & \widehat{\mathbf{B}} = \mathbf{B}\downarrow\mathbf{B} \\ {\scriptstyle F}\downarrow & & \downarrow{\scriptstyle \widehat{F}} \\ \mathbf{D} & \xrightarrow{\blacktriangle} & \widehat{\mathbf{D}} = \mathbf{D}\downarrow\mathbf{D} \end{array} \tag{1.12}$$

and its diagonal arrow is equal to the nat-functor N_τ *defined by* (1.9).

Proof. Let us show the commutativity of the diagram above:

1. For any object a of $\mathbf{B}$, we have that

$$\blacktriangle^0 F^0(a) = J\tau_I(F^0(a)) = J(\mathrm{id}_{F^0(a)}) = J(F^1(\mathrm{id}_a)), \quad \text{and}$$

$$\widehat{F}^0\blacktriangle^0(a) = \widehat{F}^0(J(\mathrm{id}_a)) = JF^1\psi J(\mathrm{id}_a) = J(F^1(\mathrm{id}_a)), \quad \text{so that}$$
$$\blacktriangle^0 F^0 = \widehat{F}^0\blacktriangle^0.$$

2. For any arrow $f : a \to b$,

$$\blacktriangle^1 F^1(f) = (F^1(f); F^1(f)) \quad \text{and}$$
$$\widehat{F}^1\blacktriangle^1(f) = \widehat{F}^1(f;f) = (F^1F^1_{\mathrm{st}}; F^1S^1_{\mathrm{nd}})(f;f) = (F^1(f); F^1(f)), \quad \text{so that}$$
$$\blacktriangle^1 F^1 = \widehat{F}^1\blacktriangle^1.$$

Consequently, we obtain the commutativity $\widehat{F} \circ \blacktriangle = \blacktriangle \circ F$ in **Cat**. □

The next proposition is important for the relationship between n-dimensional levels and symmetrical categories (for the conceptually closed categories defined in Chapter 3).

Proposition 2. *For each n-dimensional level* $\mathbf{C}_n$, $n \geq 2$, *with its identity endofunctor* $\mathrm{Id}_{\mathbf{C}_n}$, *there exist the natural transformations* $\sigma : \blacktriangle \circ F_{\mathrm{st}} \xrightarrow{\cdot} \mathrm{Id}_{\mathbf{C}_n}$ *and* $\sigma^{-1} : \mathrm{Id}_{\mathbf{C}_n} \xrightarrow{\cdot} \blacktriangle \circ S_{\mathrm{nd}}$, *such that*[1]

$$\sigma = (\psi\blacktriangle^0 F^0_{\mathrm{st}}; \psi), \quad \textit{and} \quad \sigma^{-1} = (\psi; \psi\blacktriangle^0 S^0_{\mathrm{nd}}). \tag{1.13}$$

The vertical composition $\sigma^{-1} \bullet \sigma = (\psi;\psi)$ *is the couple composed by the fundamental natural transformation* (1.5), *i. e., its function* J^{-1} *in* (1.2).

Proof. Directly from the properties of the diagonal functor $\blacktriangle$ in (1.11), with its object component $\blacktriangle^0 = J\tau_I$, by considering the following scheme in $\mathbf{C}_n$ for natural transformation σ, for two arrows $f : a \to b$ and $g : c \to d$:

$$\begin{array}{ccccc}
J(f) & & \blacktriangle^0 F^0_{\mathrm{st}}(J(f)) = J(\mathrm{id}_a) & \xrightarrow{\sigma(J(f)) = (\mathrm{id}_a; f)} & \mathrm{Id}^0_{\mathbf{C}_n}(J(f)) = J(f) \\
\Big\downarrow (h;k) & & \Big\downarrow \blacktriangle^1 F^1_{\mathrm{st}}(h;k) = (h;h) & & \Big\downarrow \mathrm{Id}^1_{\mathbf{C}_n}(h;k) = (h;k) \\
J(g) & & \blacktriangle^0 F^0_{\mathrm{st}}(J(g)) = J(\mathrm{id}_c) & \xrightarrow{\sigma(J(g)) = (\mathrm{id}_c; g)} & \mathrm{Id}^0_{\mathbf{C}_n}(J(g)) = J(g)
\end{array} \tag{1.14}$$

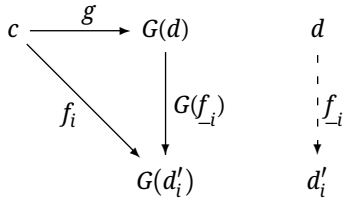
with $\sigma(J(f)) \triangleq (\mathrm{id}_{\mathrm{dom}(f)}; f)$. Analogously, the arrow components of the natural transformation σ^{-1} are computed in a standard way by its functional representation for each object $J(f)$, by $\sigma^{-1}(J(f)) = (f; \mathrm{id}_{\mathrm{cod}(f)})$. It is easy to verify that this function in (1.13) is well-defined.

So, for the vertical composition we have that $(\sigma^{-1} \bullet \sigma)(J(f)) = \sigma^{-1}(J(f)) \circ \sigma(J(f)) = (\mathrm{id}_{\mathrm{cod}(f)}; f) \circ (f; \mathrm{id}_{\mathrm{dom}(f)}) = (f;f) = (\psi(J(f)); \psi(J(f))) = (\psi;\psi)(J(f))$, i. e., $\sigma^{-1} \bullet \sigma = (\psi;\psi)$. □

1 We recall that for the simplicity we consider the natural transformations as a function from objects into arrows that are components of this natural transformation.

The next theorem demonstrates one of the fundamental structural properties preserved by the comma lifting.

Theorem 1. *Given a functor* $G : \mathbf{D} \to \mathbf{C}$*, for each universal arrow* (d, g)*, with* $d \in \mathrm{Ob}_{\mathbf{D}}$ *and* $g : c \to G(d)$ *in* $\mathbf{C}$*, from* c *to functor* G*, presented by scheme* (A.5)[2] *in the Appendix,*

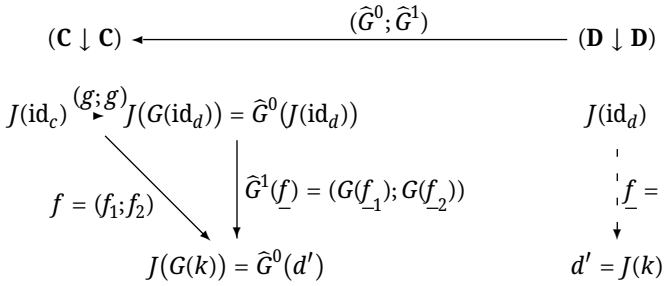


(1.15)

there is the universal arrow $(J(\mathrm{id}_d), (g, g))$ *from* $J(\mathrm{id}_d)$ *to the comma-lifted functor* $\widehat{G} : (\mathbf{D} \downarrow \mathbf{D}) \to (\mathbf{C} \downarrow \mathbf{C})$*. If* $F : \mathbf{C} \to \mathbf{D}$ *is left-adjoint functor to* G*, with adjunction* $(F, G, \varepsilon, \eta)$ *with natural transformations, unit* $\eta : \mathrm{Id}_{\mathbf{C}} \xrightarrow{\cdot} GF$ *and counit* $\varepsilon : FG \xrightarrow{\cdot} \mathrm{Id}_{\mathbf{D}}$ *then there exists the comma-lifted adjunction* $(\widehat{F}, \widehat{G}, \widehat{\varepsilon}, \widehat{\eta})$*.*

Proof. For the universal arrow (d, g) such that for each pair (d'_i, f_i), there is the unique arrow $\underline{f} : d \to d'_i$ such that the diagram in (1.15) commutes, we have the corresponding diagram for the arrows in the comma-lifted category $(\mathbf{C} \downarrow \mathbf{C})$, for the comma-lifted universal arrow $(J(\mathrm{id}_d), (g, g))$; so that for each pair (d', f) where $d' = J(k)$ for some arrow $k : d'_1 \to d'_2$ in $\mathbf{D}$ and $f = (f_1; f_2) : J(\mathrm{id}_c) \to \widehat{G}(d')$, and hence from the fact that $\widehat{G}(d') = (JG\psi)(J(k)) = J(G(k))$ for this arrow f it must hold that

$$f_2 = G(k) \circ f_1 \tag{1.16}$$



(1.17)

From the fact that (d, g) is an universal arrow, for the pairs (d'_1, f_1) and (d'_2, f_2), the following diagrams in $\mathbf{C}$ commute:

$$f_1 = G(\underline{f}_{_1}) \circ g, \quad \text{and} \quad f_2 = G(\underline{f}_{_2}) \circ g. \tag{1.18}$$

2 Where for each pair (d'_i, f_i), there exists *the unique* arrow $\underline{f}_{_i} : d \to d'_i$ such that $f_i = G(\underline{f}_{_i}) \circ g$.

It is easy now to verify that $(J(\mathrm{id}_d),(g,g))$ is indeed a comma-lifted universal arrow obtained from the universal arrow (d,g), so that there exists the unique arrow $\underline{f} = (\underline{f}_{-1};\underline{f}_{-2}) : J(\mathrm{id}_d) \to d'$ for any pair (d',f), composed by the unique arrows $\underline{f}_{-1}$ and $\underline{f}_{-2}$ for the pairs (d'_1,f_1) and (d'_2,f_2) relatively, with

$$\underline{f}_{-2} = k \circ \underline{f}_{-1}, \quad \text{and hence} \quad G(\underline{f}_{-2}) = G(k) \circ G(\underline{f}_{-1}) \tag{1.19}$$

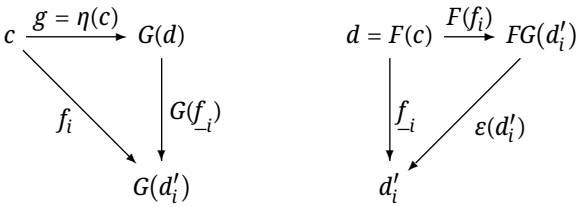
such that the diagram above in the comma-lifted category $\widehat{\mathbf{C}} = \mathbf{C} \downarrow \mathbf{C}$ commutes, i. e., $f = \widehat{G}(\underline{f}) \circ (g;g)$, i. e.,

$$(f_1;f_2) = \big(G(\underline{f}_{-1});G(\underline{f}_{-2})\big) \circ (g;g) = \big(G(\underline{f}_{-1}) \circ g; G(\underline{f}_{-2}) \circ g\big)$$

in accordance with (1.18).

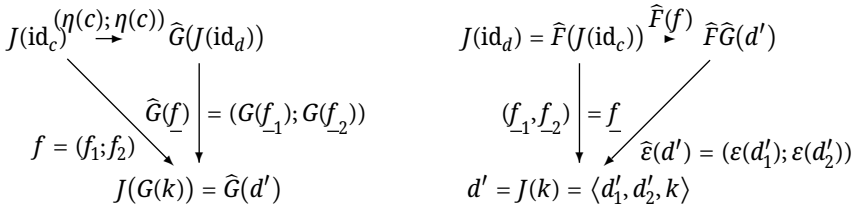
Thus, the $(J(\mathrm{id}_d),(g;g))$ is the comma-lifted universal arrow in the higher $(n+1)$-dimensional level obtained from the universal arrow (d,g) in the lower n-dimensional level.

In the case of the adjunction, the diagram 1.15 extends to these two commutative diagrams



(1.20)

so that the comma-lifted adjunction is represented by the following commutative diagrams, with $\widehat{\eta}(J(\mathrm{id}_c)) = (\eta(c);\eta(c))$, which extends that in (1.21):



(1.21)

with $\widehat{F}(f) = (F(f_1);F(f_2))$ and $\widehat{F}\widehat{G}(d') = \widehat{F}\widehat{G}(J(k)) = J(FG(k)) = \langle FK(d'_1),FK(d'_2),FG(k)\rangle$.

□

1.3 Introduction to comma-induction in the n-dimensional hierarchy

This section, dedicated to comma-induction, which is based on the previously introduced operations of comma lifting, is a short introduction to the more general phenomena of the comma-propagation, which will be discussed in Section 2.1. Just because of that, some proofs will be omitted (they can be found in [4]) because they will be provided for the more general framework for comma-propagation in the next chapter.

The phenomena of the comma-induction, in the n-dimensional hierarchy, for the functors and their derived concepts as natural transformations and adjunctions, is an analog to the effects of the harmonics in the given spectra (like the spectra of the sounds, waves, etc.). The existence of a functor between two basic categories (1-dimensional level) causes the existence of the comma-lifted functors between the arrow categories of higher n-dimensional levels, for $n \geq 1$. In the pure comma-induction, the comma-induced functor at n-dimensional level of the base functor (in the 0-dimensional level) is just the iteration of the comma lifting $n \geq 1$ times.

So, depending on the domain and codomain base categories of a given functor (between an n-dimensional level into an m-dimensional level), such a comma-induction into higher levels is denominated as "'ascending" if $n < m$, as "descending" if $n > m$ or "balanced" if $n = m$. The most interesting are the comma-induced functors when $|n - m| \leq 1$, defined as follows.

Definition 3. We define the following four cases of the comma-induction of a given base covariant functor F (considered as 1-dimensional level), which for each n-dimensional level, $n \geq 2$, will be denoted by F_n comma-induced functor, as follows:

1. Basic case: when the base functor is $F : \mathbf{D} \to \mathbf{C}$, with $F_1 \triangleq F$, and let $F_{n-1} : \mathbf{D}_{n-1} \to \mathbf{C}_{n-1}$ be a $(n-1)$-dimensional comma-induced functor. Then the n-dimensional comma-induced functor is its comma lifted,

$$F_n \triangleq \widehat{F}_{n-1} : \mathbf{D}_n \to \mathbf{C}_n.$$

2. Descending case: when the base functor is $F : (\mathbf{D} \downarrow \mathbf{D}) \to \mathbf{C}$, with $F_1 \triangleq F$, and let $F_{n-1} : \mathbf{D}_n \to \mathbf{C}_{n-1}$ be a $(n-1)$-dimensional comma-induced functor. Then the n-dimensional comma-induced functor is

$$F_n \triangleq \widehat{F}_{n-1} \circ L_\perp : \mathbf{D}_{n+1} \to \mathbf{C}_n.$$

3. Ascending case: when the base functor is $F : \mathbf{D} \to (\mathbf{C} \downarrow \mathbf{C})$, with $F_1 \triangleq F$, and let $F_{n-1} : \mathbf{D}_{n-1} \to \mathbf{C}_n$ be $(n-1)$-dimensional comma-induced functor. Then n-dimensional comma-induced functor is

$$F_n \triangleq L_\perp \circ \widehat{F}_{n-1} : \mathbf{D}_n \to \mathbf{C}_{n+1}.$$

4. Balanced case: when the base functor is $F : (\mathbf{D} \downarrow \mathbf{D}) \to (\mathbf{C} \downarrow \mathbf{C})$, with $F_1 \triangleq F$, and let $F_{n-1} : \mathbf{D}_n \to \mathbf{C}_n$ be a $(n-1)$-dimensional comma-induced functor. Then the n-dimensional comma-induced functor is

$$F_n \triangleq L_\perp \circ \widehat{F}_{n-1} \circ L_\perp : \mathbf{D}_{n+1} \to \mathbf{C}_{n+1}.$$

In the case when $\mathbf{D} = \mathbf{C}$, we obtain the *strict* comma-induction.

So, the comma-induction in the basic case corresponds to the iteration of the comma lifting $n \geq 1$ times, while for another cases this iteration is modulated by using the "duality" endofunctor $L_\perp = (\Psi, \Phi)$ in (1.10), which is a specific functor modulator, which is used for a general comma-propagation as will be explained in Section 2.1.

Based on the Corollary 1, the comma-induction, e. g., in the basic case of functors $F : \mathbf{D} \to \mathbf{C}$ with $\mathbf{D}_1 = \mathbf{D}$ and $F_1 = F$, can be represented by an infinite chain of the n-dimensional levels commutative diagram in category **Cat**:

$$\begin{array}{ccccccc}
\mathbf{D}_1 & \blacktriangle & \to & \mathbf{D}_2 & \blacktriangle \to \ \ldots & \mathbf{D}_n & \blacktriangle \to \ \ldots \\
\downarrow F_1 & & & \downarrow F_2 & \ \ldots & \downarrow F_n & \\
\mathbf{C}_1 & \blacktriangle & \to & \mathbf{C}_2 & \blacktriangle \to \ \ldots & \mathbf{C}_n & \blacktriangle \to \ \ldots
\end{array} \tag{1.22}$$

The comma-induction can be applied also to the natural transformations.

Definition 4. We define the following four cases of the comma-induction of a given base natural transformation $\tau : F \xrightarrow{\cdot} G$ (considered as 1-dimensional level denoted by τ_1), which for each $(n-1)$-dimensional level, $n \geq 2$, will be denoted by $\tau_{n-1} : F_{n-1} \xrightarrow{\cdot} G_{n-1}$ comma-induced natural transformation, as follows:

1. Basic case: when the base functors are $F, G : \mathbf{D} \to \mathbf{C}$. Then the n-dimensional comma-induced natural transformation is its comma lifted (from (1.8)),

$$\tau_n \triangleq \widehat{\tau}_{n-1} : \widehat{F}_{n-1} \xrightarrow{\cdot} \widehat{G}_{n-1},$$

   that is the function $\tau_n \triangleq \widehat{\tau}_{n-1} = (\tau_{n-1} F^0_{\mathrm{st}}; \tau_{n-1} S^0_{\mathrm{nd}})$.
2. Descending case: when the base functors are $F, G : (\mathbf{D} \downarrow \mathbf{D}) \to \mathbf{C}$. Then the n-dimensional comma-induced natural transformation is (from (A.9) in the Appendix)

$$\tau_n \triangleq \widehat{\tau}_{n-1} L_\perp : \widehat{F}_{n-1} \circ L_\perp \xrightarrow{\cdot} \widehat{G}_{n-1} \circ L_\perp,$$

   that is the function $\tau_n \triangleq (\tau_{n-1} F^0_{\mathrm{st}}; \tau_{n-1} S^0_{\mathrm{nd}}) L^0_\perp$.
3. Ascending case: when the base functor is $F, G : \mathbf{D} \to (\mathbf{C} \downarrow \mathbf{C})$. Then n-dimensional comma-induced natural transformation is (from (A.10) in the Appendix)

$$\tau_n \triangleq L_\perp \widehat{\tau}_{n-1} : L_\perp \circ \widehat{F}_{n-1} \xrightarrow{\cdot} L_\perp \circ \widehat{G}_{n-1},$$

that is the function $\tau_n \triangleq L^1_\perp(\tau_{n-1}F^0_{\mathrm{st}}; \tau_{n-1}S^0_{\mathrm{nd}})$.

4. Balanced case: when the base functors are $F, G : (\mathbf{D} \downarrow \mathbf{D}) \to (\mathbf{C} \downarrow \mathbf{C})$ Then n-dimensional comma-induced natural transformation is

$$\tau_n \triangleq L_\perp \widehat{\tau}_{n-1} L_\perp : L_\perp \circ \widehat{F}_{n-1} \circ L_\perp \xrightarrow{\cdot} L_\perp \circ \widehat{G}_{n-1} \circ L_\perp,$$

that is the function $\tau_n \triangleq L^1_\perp(\tau_{n-1}F^0_{\mathrm{st}}; \tau_{n-1}S^0_{\mathrm{nd}})L^0_\perp$,

where $L^0_\perp = \Psi$ and $L^1_\perp = \Phi$ are the object's and arrows's components of the duality endofunctor $L_\perp$ in (1.10).

So, as in the case of the comma-induced functors, also for the comma-induced natural transformations different from basic case, we have the modulation by the components of the duality endofunctor $L_\perp$. It is easy to verify that in all four cases above, from Definition 3, we obtain that for each $n \geq 1$, the comma-induced natural transformation is just between comma-induced functor derived from the base functors F and G:

$$\tau_n : F_n \xrightarrow{\cdot} G_n$$

It is shown in [4] that the following distributive laws for the operation of comma-induction '$(_)_n$' are satisfied:

1. For the operation of composition of functors:

$$F_n \circ G_n = (F \circ G)_n$$

2. For the horizontal composition of natural transformations:

$$\tau_n \circ \eta_n = (\tau \circ \eta)_n$$

3. For the vertical composition of natural transformations:

$$\tau_n \bullet \eta_n = (\tau \bullet \eta)_n.$$

There exists the following closure property for the *strict* comma-induction.

Corollary 2. *The following property of closure for strict comma-induction are valid:*

1. *Let F be a base covariant functor in Definition* 3*. Any functor induced from a comma-induced functor F_n, for $n \geq 1$, is a comma-induced functor from F (with $F_1 = F$). We denote by $\mathfrak{R}_I(F_n) \triangleq \{F_m \mid m \geq n\}$ the set of all comma-induced functors from the functor F_n. Then $\mathfrak{R}_I$ is an operation such that for any $n \geq 1$,*

$$\mathfrak{R}_I(F) \supseteq \mathfrak{R}_I(F_n) = \mathfrak{R}_I(F_{n+1}) \bigcup \{F_n\}.$$

2. *Let τ be a natural transformation. Any natural transformation induced from a comma-induced natural transformation τ_n, for $n \geq 1$, is a comma-induced natural transformation from τ (with $\tau_1 = \tau$).*
We denote by $\mathfrak{R}_{IT}(\tau_n) \triangleq \{\tau_m \mid m \geq n\}$ the set of all comma-induced natural transformations from τ_n. Then $\mathfrak{R}_{IT}$ is an operation such that for any $n \geq 1$,

$$\mathfrak{R}_{IT}(\tau) \supseteq \mathfrak{R}_{IT}(\tau_n) = \mathfrak{R}_{IT}(\tau_{n+1}) \bigcup \{\tau_n\}.$$

Proof. By induction (Theorems 2 and 3 in [4]). □

Let us present the following example of the theory developed previously, which will be useful for the Conceptually Closed Categories (CoCC), which have important properties of the category symmetry theory provided in Section 3.3, Chapter 3.

Example 3. Let us consider for a given base category $\mathbf{C}$ with its identity functor $I_{\mathbf{C}} : \mathbf{C} \to \mathbf{C}$ and a descending functor $T_e : \mathbf{C} \downarrow \mathbf{C} \to \mathbf{C}$, the two natural transformations,

$$\varphi : T_e \circ \blacktriangle \xrightarrow{\cdot} I_{\mathbf{C}} \quad \text{and} \quad \varphi^{-1} : I_{\mathbf{C}} \xrightarrow{\cdot} T_e \circ \blacktriangle \tag{1.23}$$

shown in the scheme

$$\begin{array}{ccc} & \xrightarrow{\quad I_{\mathbf{C}} \quad} & \\ & \downarrow \varphi^{-1} & \\ \mathbf{C} & \xrightarrow{\quad T_e \circ \blacktriangle \quad} & \mathbf{C} \\ & \downarrow \varphi & \\ & \xrightarrow[\quad I_{\mathbf{C}} \quad]{} & \end{array} \tag{1.24}$$

with the two (nonidentity) vertical compositions $\varphi^{-1} \bullet \varphi : T_e \circ \blacktriangle \xrightarrow{\cdot} T_e \circ \blacktriangle$ and $\varphi \bullet \varphi^{-1} : I_{\mathbf{C}} \xrightarrow{\cdot} I_{\mathbf{C}}$.

So, let us show the comma-induction. Let us, for a given *n*-dimensional level $\mathbf{C}_n$ with comma-induced functor $Te_n : \mathbf{C}_n \downarrow \mathbf{C}_n \to \mathbf{C}_n$ and an arrow $f : a \to b$, consider the following commutative diagrams for these two vertical composition of natural transformations:

$$\begin{array}{ccccccc}
a & a = I^0_{\mathbf{C}_n}(a) & \xrightarrow{\varphi_n^{-1}(a)} & J(Te^1_n(\mathrm{id}_a)) = Te^0_n \blacktriangle^0_n(a) & \xrightarrow{\varphi_n(a)} & a = I^0_{\mathbf{C}_n}(a) \\
\Big\downarrow f & \Big\downarrow f = I^1_{\mathbf{C}_n}(f) & & \Big\downarrow Te^1_n(f;f) = Te^1_n \blacktriangle^1(f) & & \Big\downarrow f = I^1_{\mathbf{C}_n}(f) \\
b & b = I^0_{\mathbf{C}_n}(b) & \xrightarrow{\varphi_n^{-1}(b)} & J(Te^1_n(\mathrm{id}_b)) = Te^0_n \blacktriangle^0_n(b) & \xrightarrow{\varphi_n(b)} & b = I^0_{\mathbf{C}_n}(a) \\
\text{in } \mathbf{C}_n & I_{\mathbf{C}_n} & \xrightarrow{\varphi_n^{-1}} & Te_n \circ \blacktriangle_n & \xrightarrow{\varphi_n} & I_{\mathbf{C}_n}
\end{array}$$

Thus, by comma-induction we obtain for the $(n+1)$-dimensional level $\mathbf{C}_{n+1}$ and an its arrow $(h;k) : J(f) \to J(g)$, corresponding to the equation $g \circ h = k \circ f$ (commutative

diagram in $\mathbf{C}_n$), we obtain the following commutative diagrams for these two vertical composition of natural transformations:

$$\begin{array}{ccccc}
J(f) & J(f)=I^0_{\mathbf{C}_{n+1}}(J(f)) & \xrightarrow{\varphi^{-1}_{n+1}(J(f))} & Te^0_{n+1}\blacktriangle^0_{n+1}(J(f)) & \xrightarrow{\varphi_{n+1}(J(f))} & J(f)=I^0_{\mathbf{C}_{n+1}}(J(f)) \\
\downarrow{\scriptstyle (h;k)} & \downarrow{\scriptstyle (h;k)=I^1_{\mathbf{C}_{n+1}}(h;k)} & & \downarrow{\scriptstyle Te^1_{n+1}\blacktriangle^1_{n+1}(h;k)} & & \downarrow{\scriptstyle (h;k)=I^1_{\mathbf{C}_{n+1}}(h;k)} \\
J(g) & J(g)=I^0_{\mathbf{C}_{n+1}}(J(g)) & \xrightarrow{\varphi^{-1}_{n+1}(J(g))} & Te^0_{n+1}\blacktriangle^0_{n+1}(J(g)) & \xrightarrow{\varphi_{n+1}(J(g))} & J(g)=I^0_{\mathbf{C}_{n+1}}(J(g))
\end{array}$$

$$\text{in } \mathbf{C}_{n+1} \qquad I_{\mathbf{C}_{n+}} \xrightarrow{\varphi^{-1}_{n+1}} Te_{n+1}\circ\blacktriangle_{n+1} \xrightarrow{\varphi_{n+1}} I_{\mathbf{C}_{n+1}}$$

where, for the central vertical arrow of the commutative diagram above, we obtain

$$\begin{aligned}
Te^1_{n+1}\blacktriangle^1_{n+1}(h;k) &= (\widehat{Te_n}L_{\perp})(L_{\perp}\blacktriangle_n)(h;k) \\
&= \widehat{Te_n}\widehat{\blacktriangle_n}(h;k) = \widehat{Te_n}(\blacktriangle^1_n F^1_{\mathrm{st}};\blacktriangle^1_n S^1_{\mathrm{nd}})(h;k) \\
&= \widehat{Te_n}(\blacktriangle^1_n(h);\blacktriangle^1_n(k)) = \widehat{Te_n}((h;h);(k;k)) \\
&= (Te^1_n(h;h);Te^1_n(k;k))
\end{aligned}$$

and

$$\begin{aligned}
Te^0_{n+1}\blacktriangle^0_{n+1}(J(f)) &= \widehat{Te_n}\widehat{\blacktriangle_n}(J(f)) \\
&= (JTe^1_n\psi)(J\blacktriangle^1_n\psi)(J(f)) = JTe^1_n\blacktriangle^1_n(f) \\
&= J(Te^1_n(f;f)).
\end{aligned}$$

Hence, this vertical arrow in $\mathbf{C}_{n+1}$ corresponds to the following commutative diagram in $\mathbf{C}_n$ with $Te_n : \mathbf{C}_{n+1} \to \mathbf{C}_n$:

$$\begin{array}{ccc}
Te^0_n(J(\mathrm{id}_a)) & \xrightarrow{Te^1_n(f;f)} & Te^0_n(J(\mathrm{id}_b)) \\
\downarrow{\scriptstyle Te^1_n(h;h)} & & \downarrow{\scriptstyle Te^1_n(k;k)} \\
Te^0_n(J(\mathrm{id}_c)) & \xrightarrow{Te^1_n(g;g)} & Te^0_n(J(\mathrm{id}_d))
\end{array}$$

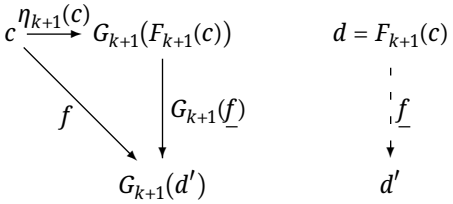
1.4 Comma-induction of the adjunctions

Based on these comma-induction properties, for covariant functors and natural transformations, the following theorem demonstrates that the comma-induction preserves all adjunctions between n-dimensional levels. The proof of this theorem is the first time provided in [4], in slightly different manner than here. Its long proof explains how in a

constructive inductive way we generate the adjunctions of the $(k+1)$-dimensional level, based on the adjunctions in the previous k-dimensional level.

Theorem 2. *If a functor F is left adjoint of a functor G with η and ε the unit and counit of this adjunction $(F, G, \varepsilon, \eta)$, then every pair of comma-induced functors F_n and G_n, for $n \geq 0$, creates an adjunction with comma-induced unit η_n and counit ε_n.*

Proof. Let $(F, G, \varepsilon, \eta)$ be the adjunction in Definition 58 in Section A.2, with counit $\varepsilon : FG \xrightarrow{\cdot} \mathrm{Id}_{\mathrm{dom}(G)}$ and unit $\eta : \mathrm{Id}_{\mathrm{dom}(F)} \xrightarrow{\cdot} GF$ and with the commutative diagrams in A.16. Let us assume by induction that we have the adjunction $(F_n, G_n, \varepsilon_n, \eta_n)$ for $n = k$. We have to show that this adjunction there exists for $n = k+1$ as well, with comma-induced natural transformation $\eta_{k+1} : \mathrm{Id}_{\mathrm{dom}(F_{k+1})} \xrightarrow{\cdot} G_{k+1}F_{k+1}$ (and analogously for the counit ε), such that for any object c we obtain the universal arrow $(F_{k+1}(c), \eta_{k+1}(c))$, from c to G_{k+1}. In order to demonstrate that $(F_{k+1}(c), \eta_{k+1}(c))$ is an universal arrow, we must prove that for each pair (d', f) with object d' and arrow $f : c \to G_{k+1}(d')$, there exists a unique arrow $\underline{f} : F_{k+1}(c) \to d'$ such that the following diagram, from A.16 and 1.15, commutes



(1.25)

Depending on the property of the functors w. r. t. the comma-induction (if are basic, ascending, descending or balanced as defined in Definition 3), we will have the following cases (in each of them we need also to control which case is valid for the unit (comma-induction of natural transformations in Definition 4) as well):

Case 1. When $F : \mathbf{C} \to \mathbf{D}$ is basic, so that $F_{k+1} = \widehat{F}_k$, and $G : \mathbf{D} \to \mathbf{C}$ is basic, and hence $G_{k+1} = \widehat{G}_k$ and η is basic case (between two endofunctors of the category $\mathbf{C}$), and hence $\eta_{k+1} = \widehat{\eta}_k$. So, we want to show that for the comma-induced $F_{k+1} : \mathbf{C}_{k+1} \to \mathbf{D}_{k+1}$ and $G_{k+1} : \mathbf{D}_{k+1} \to \mathbf{C}_{k+1}$, where the natural transformation η_{k+1} is the unit of their adjunction.

Let $c = J(h)$ of the arrow $h : a \to b$ in $\mathbf{C}_k$ and $d' = J(g)$ of the arrow $g : a_1 \to b_1$ in $\mathbf{D}_k$. So, from the fact that $\eta_{k+1} = \widehat{\eta}_k$ is the comma lifting of the natural transformation η_k, and $F_{k+1} = \widehat{F}_k$ and $G_{k+1} = \widehat{G}_k$, we obtain that the top arrow in diagram above,

$$\begin{aligned}
\eta_{k+1}(c) &= (\eta_k F^0_{\mathrm{st}}; \eta_k S^0_{\mathrm{nd}})(J(h)) = (\eta_k(a); \eta_k(b)) \quad \text{and} \\
G_{k+1}(F_{k+1}(c)) &= G_{k+1}(JF^1_k\psi)(J(h)) = G_{k+1}(J(F_k(h))) \\
&= J(G_kF_k(h)) = \langle G_kF_k(a), G_kF_k(b), G_kF_k(h)\rangle,
\end{aligned}$$

the arrow $\eta_{k+1}(c)$ is just the following commutative diagram of the natural transformation $\eta_k : \mathrm{Id}_{\mathbf{C}_k} \xrightarrow{\cdot} G_k F_k$ (which is the unit from the inductive hypothesis):

$$\begin{array}{ccc} a & \xrightarrow{\eta_k(a)} & G_k F_k(a) \\ \Big\downarrow{\scriptstyle h} & & \Big\downarrow{\scriptstyle G_k F_k(h)} \\ b & \xrightarrow{\eta_k(b)} & G_k F_k(b) \end{array}$$

From the fact that

$$\begin{aligned} G_{k+1}(d') &= \widehat{G}_k(J(g)) = (JG_k^1\psi)(J(g)) = J(G_k(g)) \\ &= \langle G_k(a_1), G_k(b_1), G_k(g)\rangle, \end{aligned}$$

and the arrow $\underline{f} = (\underline{f}_{-1};\underline{f}_{-2}) : d \to d'$, i. e., $\underline{f} = (\underline{f}_{-1};\underline{f}_{-2}) : J(F_k(h)) \to J(g)$ is a commutative diagram $g \circ \underline{f}_{-1} = \underline{f}_{-2} \circ F_k(h)$ in $\mathbf{D}_k$, we obtain that

$$G_{k+1}(\underline{f}) = \widehat{G}_k(\underline{f}_{-1};\underline{f}_{-2}) = (G_k^1 F_{\mathrm{st}}^1; G_k^1 S_{\mathrm{nd}}^1)(\underline{f}_{-1};\underline{f}_{-2}) = (G_k(\underline{f}_{-1}); G_k(\underline{f}_{-2})).$$

Thus, the arrow $f = (f_1;f_2) : c \to G_{k+1}(d')$ is equal to $f = (f_1;f_2) : J(h) \to J(G_k(g))$, and hence represents the commutative diagram $G_k(g) \circ f_1 = f_2 \circ h$ in $\mathbf{D}_k$, with the arrows

$$\underline{f}_{-1} : F_k(a) \to a_1 \quad \text{and} \quad \underline{f}_{-2} : F_k(b) \to b_1. \tag{1.26}$$

Consequently, the commutativity of the diagram (1.25) in $\mathbf{C}_{k+1}$, $f = G_{k+1}(\underline{f}) \circ \eta_{k+1}(c)$ is represented by two commutative diagrams in $\mathbf{C}_k$:

$$f_1 = G_k(\underline{f}_{-1}) \circ \eta_k(a) : a \to G_k(a_1) \quad \text{and} \quad f_2 = G_k(\underline{f}_{-2}) \circ \eta_k(b) : b \to G_k(b_1). \tag{1.27}$$

So, from the inductive hypothesis for the k-level adjunction $(F_k, G_k, \varepsilon_k, \eta_k)$, we have that the pair $(F_k(a), \eta_k(a))$ is an universal arrow, such that for each pair (a_1, f_1) there exists a *unique* arrow $\underline{f}_{-1}$ in (1.26) such that the first commutativity (equation) in (1.27) is valid. In the same way, we have that the pair $(F_k(b), \eta_k(b))$ is an universal arrow, such that for each pair (b_1, f_2) there exists a *unique* arrow $\underline{f}_{-2}$ in (1.26) such that the second commutativity (equation) in (1.27) is valid as well.

Consequently, for the arrow $\underline{f} = (\underline{f}_{-1};\underline{f}_{-2}) : d \to d'$, for which we have shown above that the diagram in (1.25) commutes for a given pair (d', f), we obtain that (d, η_{k+1}) is the universal arrow, and hence η_{k+1} is the unit. In an analog way, we can show that the induced natural transformation ε_{k+1} (by using the couniversal arrows) is the counit, which compose the $(k+1)$-level induced adjunction $(F_{k+1}, G_{k+1}, \varepsilon_{k+1}, \eta_{k+1})$.

Case 2. When $F : \mathbf{C} \to (\mathbf{D} \downarrow \mathbf{D})$ is ascending, so that $F_{k+1} = L_\perp \widehat{F}_k$, and $G : (\mathbf{D} \downarrow \mathbf{D}) \to \mathbf{C}$ is descending, so that $G_{k+1} = \widehat{G}_k L_\perp$ and η is basic case (between two endofunctors of the

category **C**), and hence $\eta_{k+1} = \widehat{\eta}_k$. Let $c = J(h_1;h_2) = \langle J(h_1'), J(h_2'), (h_1;h_2)\rangle$ of the arrow $(h_1;h_2) : J(h_1') \to J(h_2')$ in $\mathbf{C}_k$ and

$$d' = J(g_1;g_2) = \langle J(g_1'), J(g_2'), (g_1;g_2)\rangle \tag{1.28}$$

of the arrow $(g_1;g_2) : J(g_1') \to J(g_2')$ in $\mathbf{D}_{k+1}$. So, from the fact that $\eta_{k+1} = \widehat{\eta}_k$ is the comma lifting of the natural transformation η_k, we obtain that the top arrow in diagram (1.25),

$$\begin{aligned}
\eta_{k+1}(c) &= (\eta_k F^0_{\mathrm{st}}; \eta_k S^0_{\mathrm{nd}})(J(h_1;h_2)) = (\eta_k(J(h_1')); \eta_k(J(h_2'))) \quad \text{and}\\
G_{k+1}(F_{k+1}(c)) &= \widehat{G}_k L_\perp L_\perp \widehat{F}_k(J(h_1;h_2)) = \widehat{G}_k \widehat{F}_k(J(h_1;h_2))\\
&= \widehat{G}_k(JF^1_k\psi)(J(h_1;h_2)) = \widehat{G}_k(J(F_k(h_1;h_2))) = J(G_kF_k(h_1;h_2))\\
&= \langle G_kF_k(J(h_1')), G_kF_k(J(h_2')), G_kF_k(h_1;h_2)\rangle,
\end{aligned}$$

the arrow $\eta_{k+1}(c)$ is just the following commutative diagram of the natural transformation $\eta_k : \mathrm{Id}_{\mathbf{C}_k} \xrightarrow{\cdot} G_kF_k$ (which is unit from the induction hypothesis):

$$\begin{array}{ccc}
J(h_1') & \xrightarrow{\eta_k(J(h_1'))} & G_kF_k(J(h_1'))\\
\Big\downarrow{\scriptstyle (h_1;h_2)} & & \Big\downarrow{\scriptstyle G_kF_k(h_1;h_2)}\\
J(h_2') & \xrightarrow{\eta_k(J(h_2'))} & G_kF_k(J(h_2'))
\end{array} \tag{1.29}$$

From the fact that

$$\begin{aligned}
G_{k+1}(d') &= \widehat{G}_k L_\perp(J(g_1;g_2)) = (JG^1_k\psi)(J(g_1';g_2')) = J(G_k(g_1';g_2'))\\
&= \langle G_k(J(g_1)), G_k(J(g_2)), G_k(g_1';g_2')\rangle,
\end{aligned}$$

we have that in commutative diagram (1.25), $f = (f_1;f_2)$ with $f_1 : J(h_1') \to G_k(J(g_1))$ and $f_2 : J(h_2') \to G_k(J(g_2))$ which, from the inductive, generates two universal arrows for $i = 1,2$, $(F_k(J(h_i')), \eta_k(J(h_i')))$, such that for the pairs $(J(g_i), f_i)$ there are two unique arrows f_i' and the following diagram commutes:

$$\begin{array}{ccccc}
J(h_i') & \overset{\eta_k(J(h_i'))}{\rightarrow} & G_kF_k(J(h_i')) & & F_k(J(h_i'))\\
& \searrow^{f_i} & \Big\downarrow{\scriptstyle G_k(f_i')} & & \vdots\,{\scriptstyle f_i'}\\
& & G_k(J(g_i)) & & J(g_i)
\end{array} \tag{1.30}$$

Note that by using these two arrows f_1' and f_2' we can define the arrow in the $(k+1)$-level $\underline{f} \triangleq L_\perp(f_1';f_2') : J(F_k(h_1;h_2)) \to J(g_1';g_2')$, i. e., $\underline{f} : F_{k+1}(c) \to d'$ the arrow in scheme (1.25). Let us show that for this arrow the diagram in (1.25) indeed commutes, by

$$
\begin{aligned}
f &= (f_1; f_2)\\
&= (G_k(f_1') \circ \eta_k(J(h_1')); G_k(f_2') \circ \eta_k(J(h_2'))) \quad \text{from commutativity in (1.30)}\\
&= (G_k(f_1'); G_k(f_2')) \circ (\eta_k(J(h_1')); \eta_k(J(h_2')))\\
&= (G_k(f_1'); G_k(f_2')) \circ \eta_{k+1}(c) \quad \text{from (1.29)}\\
&= ((G_k^1 F_{\mathrm{st}}^1; G_k^1 S_{\mathrm{nd}}^1)(f_1'; f_2')) \circ \eta_{k+1}(c)\\
&= \widehat{G}_k(f_1'; f_2') \circ \eta_{k+1}(c) \quad \text{from (1.7)}\\
&= (\widehat{G}_k L_\perp) L_\perp(f_1'; f_2') \circ \eta_{k+1}(c) = G_{k+1}(L_\perp(f_1'; f_2')) \circ \eta_{k+1}(c)\\
&= G_{k+1}(\underline{f}) \circ \eta_{k+1}(c).
\end{aligned}
$$

So, from this commutativity and from the fact that f_1' and f_2' are two unique arrows from the inductive hypothesis in (1.30), so that also $\underline{f}$ is a unique arrow for this commutativity, we obtained that η_{k+1} is the unit. In an analog way, we can show that the induced natural transformation ε_{k+1} (by using the couniversal arrows) is the counit, which compose the $(k+1)$-level induced adjunction $(F_{k+1}, G_{k+1}, \varepsilon_{k+1}, \eta_{k+1})$.

Case 3. When $F : (\mathbf{C} \downarrow \mathbf{C}) \to \mathbf{D}$ is descending, so that $F_{k+1} = \widehat{F}_k L_\perp$, and $G : \mathbf{D} \to (\mathbf{C} \downarrow \mathbf{C})$ is ascending, so that $G_{k+1} = L_\perp \widehat{G}_k$ and η is balanced case (between two endofunctors of the category $(\mathbf{C} \downarrow \mathbf{C})$), and hence $\eta_{k+1} = L_\perp^1 \widehat{\eta}_k L_\perp^0$.

Let $c = J(h_1; h_2) = L_\perp(J(h_1'; h_2')) = \langle L_\perp(J(h_1)), L_\perp(J(h_2)), L_\perp(h_1'; h_2')\rangle$ of the arrow $(h_1'; h_2') : J(h_1) \to J(h_2)$ in $\mathbf{C}_{k+1}$ and

$$
d' = J(g_1; g_2) = \langle J(g_1'), J(g_2'), (g_1; g_2)\rangle \tag{1.31}
$$

of the arrow $(g_1; g_2) : J(g_1') \to J(g_2')$ in $\mathbf{D}_k$. So, from the fact that $\eta_{k+1} = L_\perp^1 \widehat{\eta}_k L_\perp^0$ is the comma lifting of the natural transformation η_k, we obtain that the top arrow in diagram (1.25),

$$
\begin{aligned}
\eta_{k+1}(c) &= L_\perp^1 \widehat{\eta}_k L_\perp(L_\perp(J(h_1'; h_2'))) = L_\perp^1 \widehat{\eta}_k(J(h_1'; h_2'))\\
&= L_\perp^1(\eta_k(J(h_1)); \eta_k(J(h_2))) \quad \text{and}\\
G_{k+1}(F_{k+1}(c)) &= \widehat{G}_{k+1}\widehat{F}_k L_\perp(L_\perp(J(h_1'; h_2'))) = \widehat{G}_{k+1}\widehat{F}_k(J(h_1'; h_2'))\\
&= \widehat{G}_{k+1}(JF_k^1\psi)(J(h_1'; h_2')) = \widehat{G}_{k+1}(J(F_k(h_1'; h_2'))) = L_\perp \widehat{G}_k(J(F_k(h_1'; h_2')))\\
&= L_\perp J(G_k F_k(h_1'; h_2')) = \langle L_\perp G_k F_k(J(h_1)), L_\perp G_k F_k(J(h_2)), L_\perp G_k F_k(h_1; h_2)\rangle.
\end{aligned}
$$

From the inductive hypothesis, for the unit η_k we have the following commutative diagram of this natural transformation $\eta_k : \mathrm{Id}_{\mathbf{C}_{k+1}} \xrightarrow{\cdot} G_k F_k$ (which is unit from the induction hypothesis) in $\mathbf{C}_{k+1}$:

$$\begin{array}{ccc} J(h_1) & \xrightarrow{\eta_k(J(h_1))} & G_kF_k(J(h_1)) \\ \downarrow{\scriptstyle (h_1';h_2')} & & \downarrow{\scriptstyle G_kF_k(h_1';h_2')} \\ J(h_2) & \xrightarrow{\eta_k(J(h_2))} & G_kF_k(J(h_2)) \end{array} \tag{1.32}$$

and two universal arrows $(F_k(J(h_i)), \eta_k(J(h_i)))$, for $i = 1,2$, such that for the pairs $(J(g_i'), f_i')$ there is a unique arrow $\underline{f}_i : F_k(J(h_i)) \to J(g_i')$ such that the following diagram commutes in $\mathbf{C}_{k+1}$:

$$\begin{array}{ccc} J(h_i) & \overset{\eta_k(J(h_i))}{\blacktriangleright} G_kF_k(J(h_i)) & \qquad F_k(J(h_i)) \\ & \searrow^{f_i'} \quad \downarrow{\scriptstyle G_k(\underline{f}_i)} & \qquad \vdots\,{\scriptstyle \underline{f}_i} \\ & G_k(J(g_i')) & \qquad J(g_i') \\ & \text{in } \mathbf{C}_{k+1} & \qquad \text{in } \mathbf{D}_k \end{array} \tag{1.33}$$

So, by combining these two commutative diagrams and (1.32), we obtain the following commutative diagram in higher level $\mathbf{C}_{k+2}$:

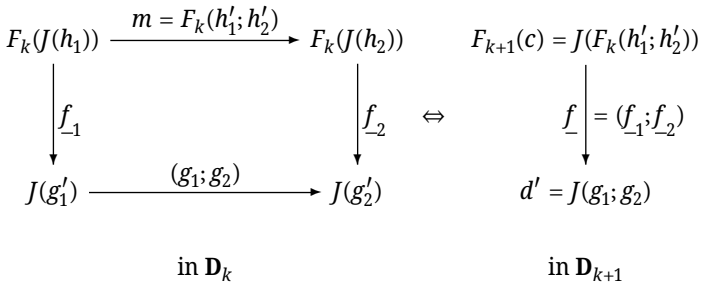
$$\begin{array}{ccc} J(h_1';h_2') & \overset{(\eta_k(a);\eta_k(b))}{\blacktriangleright} & J(G_k(m)) \\ & \searrow^{(f_1',f_2')} & \downarrow{\scriptstyle (G_k(\underline{f}_1);G_k(\underline{f}_2))} \\ & & J(G_k(g_1;g_2)) \end{array} \tag{1.34}$$

with $a = J(h_1)$, $b = J(h_2)$ and the arrow $m = F_k(h_1';h_2')$, and hence the arrow $(\eta_k(a);\eta_k(b))$ represents the commutative diagram (1.32). The vertical arrow in this commutative diagram above, $(G_k(\underline{f}_1);G_k(\underline{f}_2))$, is obtained by application of the functor G_k to the commutative diagram[3]

3 From the fact that

$$\begin{aligned} F_{k+1}(c) &= F_{k+1}\big(L_\perp J\big(h_1';h_2'\big)\big) = \hat{F}_k L_\perp\big(L_\perp J\big(h_1';h_2'\big)\big) = \hat{F}_k\big(J\big(h_1';h_2'\big)\big) = JF_k\psi\big(J\big(h_1';h_2'\big)\big) \\ &= J\big(F_k\big(h_1';h_2'\big)\big), \end{aligned}$$

the vertical arrow $\hat{f}$ is just well-defined.



Thus, by application of the functor $L_\perp$ to the commutative diagram (1.34), the vertical arrow of this commutative diagram becomes

$$L_\perp(G_k(\underline{f}_{-1}); G_k(\underline{f}_{-2})) = L_\perp(G_k^1 F_{\mathrm{st}}^1; G_k^1 S_{\mathrm{nd}}^1)(\underline{f}_{-1}; \underline{f}_{-2}) = L_\perp \widehat{G}_k((\underline{f}_{-1}; \underline{f}_{-2}),$$

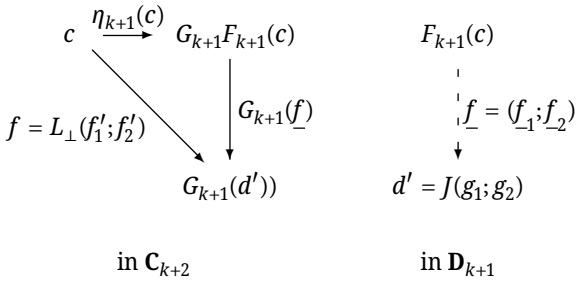
that is

$$L_\perp(G_k(\underline{f}_{-1}); G_k(\underline{f}_{-2})) = G_{k+1}(\underline{f}). \tag{1.35}$$

Moreover, we have that application of the functor $L_\perp$ to the bottom object in (1.34) is $L_\perp J(G_k(g_1; g_2)) = L_\perp J(G_k \psi J(g_1; g_2)) = L_\perp \widehat{G}_k(J(g_1; g_2))$, i. e.,

$$L_\perp J(G_k(g_1; g_2)) = G_{k+1}(J(g_1; g_2)) = G_{k+1}(d'). \tag{1.36}$$

So, the application of the functor $L_\perp$ to commutative diagram (1.34), from these two results in (1.35) and (1.36) and two previously derived results that $G_{k+1}F_{k+1}(c) = L_\perp(G_k(m))$ and $\eta_{k+1}(c) = L_\perp(\eta_k(a); \eta_k(b))$, generates the commutative diagram in the following scheme for $c = J(h_1; h_2) = L_\perp(J(h_1'; h_2'))$:

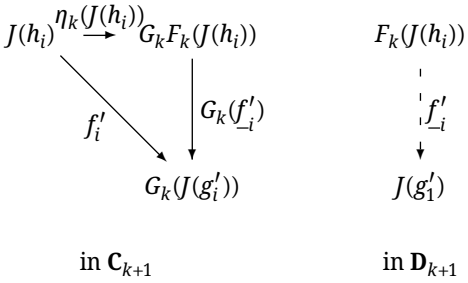


where for a pair (d', f) there is a unique arrow $\underline{f}_{-} = (\underline{f}_{-1}; \underline{f}_{-2})$ (composed by the two unique arrows in diagrams (1.34) from the inductive hypothesis of the adjunction in the k-level), such that the diagram commutes, so that the pair $(F_{k+1}(c), \eta_{k+1}(c))$ is the universal arrow, and hence we obtain that η_{k+1} is the unit. In an analog way, we can show that the induced natural transformation ε_{k+1} (by using the couniversal arrows) is the counit, which compose the $(k+1)$-level induced adjunction $(F_{k+1}, G_{k+1}, \varepsilon_{k+1}, \eta_{k+1})$.

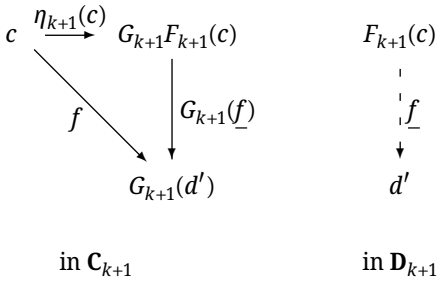
Case 4. When $F : (\mathbf{C} \downarrow \mathbf{C}) \to (\mathbf{D} \downarrow \mathbf{D})$ is balanced, so that $F_{k+1} = L_\perp \widehat{F}_k L_\perp$, and $G : (\mathbf{D} \downarrow \mathbf{D}) \to (\mathbf{C} \downarrow \mathbf{C})$ is ascending so that $G_{k+1} = L_\perp \widehat{G}_k L_\perp$, and η is a basic case (between endofunctors of $\mathbf{C}$), and hence $\eta_{k+1} = L^1_\perp \widehat{\eta}_k L^0_\perp$; so the proof is a combination of the cases 2 and 3, so (left as an exercise to readers to verify) shortly we obtain the following:

For any $c = J(h_1; h_2)$, obtained from the arrow $(h_1; h_2) : J(h'_1) \to J(h'_2)$ in $\mathbf{C}_{k+1}$, we have that the pair $(F_{k+1}(c), \eta_{k+1}(c))$ is an universal arrow. So, for each pair (d', f), where $d' = J(g_1, g_2)$ is an object in $\mathbf{D}_{k+2}$ obtained from the arrow $(g_1, g_2) : J(g'_1) \to J(g'_2)$, and an arrow $f : c \to G_{k+1}(d')$, there exists a unique arrow $\underline{f} : F_{k+1}(c) \to d'$ such that $f = G_{k+1} \circ \eta_{k+1}(c)$.

Now, let $(f'_1; f'_2) = L_\perp(f)$, from the inductive hypothesis there is the k-level adjunction $(F_k, G_k, \varepsilon_k, \eta_k)$ such that for the universal arrows $(F_k(J(h_i)), \eta_k(J(h_i)))$, for $i = 1, 2$ and the pairs $(J(g'_i), f'_i)$, there are two unique arrows $\underline{f}'_i$ such that the diagram in the following scheme commutes:



So, by combining these two commutative diagrams, and connecting them by the arrows $(h'_1, h'_2) : J(h_1) \to J(h_2)$ and $(g_1, g_2) : J(g'_1) \to J(g'_2)$, we obtain a 3-D commutative diagram, and hence by applying the functor $L_\perp$ to it, we obtain the following commutative diagram represented in the $(k+2)$-level:



where the unique arrow that makes this diagram commutative is $\underline{f} = L_\perp(\underline{f}'_1; \underline{f}'_2)$. It is easy to verify that

$$\begin{aligned}
\eta_{k+1}(c) &= L_\perp(\eta_k(J(h_1)); \eta_k(J(h_2))),\\
F_{k+1}(c) &= L_\perp(J(F_k(h'_1; h'_2))),\\
G_{k+1}F_{k+1}(c) &= L_\perp(J(G_k F_k(h'_1; h'_2))),
\end{aligned}$$

$$G_{k+1}(\underline{f}) = L_{\perp}(G_k(\underline{f}_{_1}); G_k(\underline{f}_{_1})) \quad \text{and}$$
$$G_{k+1}(d') = L_{\perp}(J(G_k(g'_1; g'_2))).$$ □

This theorem will be used for the case of imploded categories in Section 3.5.

2 Comma-propagation transformations: Global categorial symmetries

2.1 Introduction to general comma-propagation

Comma-propagation is a *general transformation* applied to n-dimensional levels, their functors and natural transformations. In what follows, the operation of the comma-propagation will be denoted by $\{_\}$, and hence applied to a category $\mathbf{C}$, functor F and natural transformation η will be denoted by $\{\mathbf{C}\}$, $\{F\}$ and $\{\eta\}$ relatively. In this section, we will introduce this operation for the n-dimensional levels by the following definition.

Definition 5 (Comma-propagation transformation of n-dimensional levels). For any given n-dimensional level $\mathbf{C}_n$, we define its comma-propagation by $\{\mathbf{C}_n\} \triangleq \mathbf{C}_{n+1}$, such that:

1. For each object c in $\mathbf{C}_n$, the comma-propagated object in $\{\mathbf{C}_n\}$ is defined by

$$\{c\} \triangleq \blacktriangle^0(c) = J(\mathrm{id}_c). \tag{2.1}$$

2. For each arrow f in $\mathbf{C}_n$, the comma-propagated arrow in $\{\mathbf{C}_n\}$ is defined by

$$\{f\} \triangleq \blacktriangle^1(f) = (f; f). \tag{2.2}$$

If we consider the universal properties of a base category as (co)universal arrows as a kind of "Lagrangian" in category theory, and derived from them adjunctions and (co)limits, then we will consider their invariance under these comma-propagation transformations: a *global symmetry* in this case means that these categorial universal structures are preserved under these general transformations *in all* n-dimensional levels.

Let us show that each $(n+1)$-dimensional level $\mathbf{C}_{n+1} = \mathbf{C}_n \downarrow \mathbf{C}_n = (\mathbf{C} \downarrow \mathbf{C})^n$, for $n \geq 1$, with $\mathbf{C}_1 = \mathbf{C}$, of a given base category $\mathbf{C}$, can be equivalently represented by the category of functors $\mathbf{C}_n^{\mathbf{J}}$ where $\mathbf{C}_n$ is n-dimensional level and the small index category $\mathbf{J}$ is equal to the preorder category $\mathbf{2}$ with two objects $a_1 = 0$ and $a_2 = 1$ and the unique nonidentity arrow $l_{12} : a_1 \to a_2$ (representing partial order $0 \leq 1$).

In fact, for an arrow (natural transformation) $\eta : F \xrightarrow{\cdot} F'$ in $\mathbf{C}_n^{\mathbf{2}}$ between two objects F and F', which are functors $F, F' : \mathbf{2} \to \mathbf{C}_n$, we have the corresponding arrow $(\eta(a_1); \eta(a_2)) : J(F(l_{12})) \to J(F'(l_{12}))$ in $\mathbf{C}_{n+1} = (\mathbf{C} \downarrow \mathbf{C})^n$, corresponding to the following commutative diagram in $\mathbf{C}_n$:

$$\begin{array}{ccc} F(a_1) & \xrightarrow{\eta(a_1)} & F'(a_1) \\ \downarrow{\scriptstyle F(l_{12})} & & \downarrow{\scriptstyle F'(l_{12})} \\ F(a_2) & \xrightarrow{\eta(a_2)} & F'(a_2) \end{array}$$

https://doi.org/10.1515/9783111081670-002

Vice versa, for each arrow $(h;k) : J(f) \to J(g)$ in $\mathbf{C}_{n+1}$, we have the arrow (natural transformation) $\eta : F \xrightarrow{\cdot} F'$ in $\mathbf{C}_n^{\mathbf{2}}$ with components $\eta(a_1) = h$ and $\eta(a_2) = k$, between functor F defined by $F(l_{12}) \triangleq f$ and functor F' defined by $F'(l_{12}) \triangleq g$. Thus, $\mathbf{C}_{n+1}$ is equivalent to $\mathbf{C}_n^{\mathbf{2}}$.

Because of that, the comma induction is discussed in the previous chapter as a particular case of functors between the category of functors $\mathbf{C}_n^{\mathbf{J}}$ and certain n-dimensional levels.

Consequently, the comma-propagated functors, natural transformation and (more generally) the concept of the comma-propagation, which we introduce in this section, generalize the comma-induced functors and natural transformations, by allowing that $\mathbf{J}$ be *any* small index category. It is well known that such categories $\mathbf{J}$ with a category of functors $\mathbf{C}^{\mathbf{J}}$ are used to define the *limits* in the base category $\mathbf{C}$, and this is the main reason for the introduction of comma-propagation, to see how the limits in this base category $\mathbf{C}_1 = \mathbf{C}$ are inductively propagated in all higher n-dimensional levels $\mathbf{C}_n$.

In fact, we can generalize the specific (for arrow categories) arrow diagonal functor $\blacktriangle : \mathbf{D} \to \mathbf{D} \downarrow \mathbf{D}$ and the standard diagonal functor $\triangle : \mathbf{D} \to \mathbf{D} \times \mathbf{D}$ and properties (A.7) in the Appendix, into a more *general* diagonal functor (ascending case) $\triangle : \mathbf{D} \to \mathbf{C}^{\mathbf{J}}$, where $\mathbf{C} = \mathbf{D}^m \triangleq \overbrace{\mathbf{D} \times \cdots \times \mathbf{D}}^{m}$ is a n-ary product of categories for finite $m \geq 1$, as follows.

Definition 6. We define the general diagonal functor $\triangle = (\triangle^0, \triangle^1) : \mathbf{D} \to \mathbf{C}^{\mathbf{J}}$, where for a finite $m \geq 1$, $\mathbf{C} = \mathbf{D}^m \triangleq \overbrace{\mathbf{D} \times \cdots \times \mathbf{D}}^{m}$ with $\mathbf{D}^1 = \mathbf{D}$, such that:

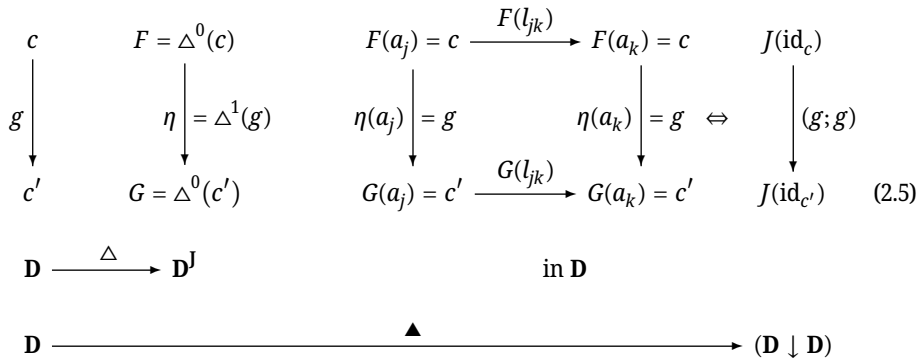
1. For each object c in $\mathbf{D}$, we define the constant functor $\triangle^0(c) : \mathbf{J} \to \mathbf{C}$, such that for all indexed objects a_j and arrows $l_{jk} : a_j \to a_k$ in $\mathbf{J}$,

$$\triangle^0(c)(a_j) = \overbrace{(c, \dots, c)}^{m} \quad \text{and} \quad \triangle^0(c)(l_{jk}) = \mathrm{id}_{(c,\dots,c)}. \tag{2.3}$$

2. For each arrow $g : c \to c'$ in $\mathbf{D}$, $\triangle^1(g)$ is a constant natural transformation, such that for each indexed object a_j in $\mathbf{J}$, the arrow component in $\mathbf{C}$ of this natural transformation is

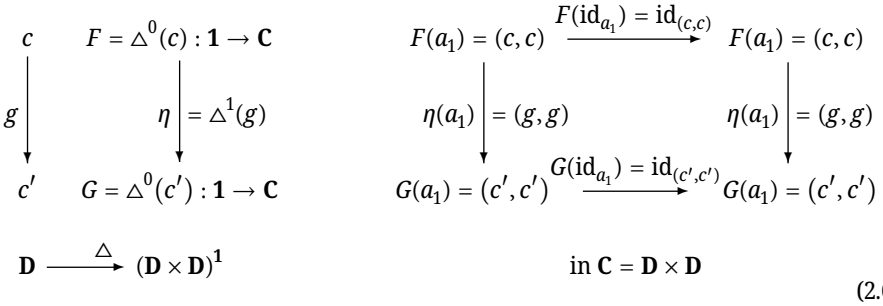
$$\triangle^1(g)(a_j) = \overbrace{(g, \dots, g)}^{m}. \tag{2.4}$$

The following scheme with the commutative diagram demonstrates the relationship between the arrow-diagonal functor $\blacktriangle : \mathbf{D} \to (\mathbf{D} \downarrow \mathbf{D})$ and this more general diagonal functor $\triangle : \mathbf{D} \to \mathbf{C}^{\mathbf{J}}$, for $m = 1$, $\mathbf{C} = \mathbf{D}^1 = \mathbf{D}$,



Note that in the case when the small index category $\mathbf{J} = \mathbf{2}$ is composed by only two objects 0 and 1 and the nonidentity arrow $e : 0 \to 1$, in the commutative diagram above (for $j = 1$ and $k = 2$) we have that $a_j = a_1 = 0$, $a_k = a_2 = 1$ and arrow $l_{jk} = l_{12} = e$, so we have the exact correspondence between the objects $\triangle^0(c)$ and $J(\mathrm{id}_c)$ and arrows $\triangle^1(g)$ and $(g;g)$ in the categories $\mathbf{D}^{\mathbf{2}}$ and $(\mathbf{D} \downarrow \mathbf{D})$, respectively.

The following scheme demonstrates the relationship between the standard diagonal functor $\triangle : \mathbf{D} \to \mathbf{D} \times \mathbf{D}$ and this more general diagonal functor $\triangle : \mathbf{D} \to \mathbf{C}^{\mathbf{J}}$, in the case when $m = 2$, $\mathbf{C} = \mathbf{D}^2 = \mathbf{D} \times \mathbf{D}$ and $\mathbf{J} = \mathbf{1}$ is the category of only one object a_1 and its identity arrow id_{a_1},



(2.6)

With this generalization, in the place of the duality functor $L_{\perp}$ in (1.10), used for the comma-induction, here for the more general comma-propagation we will introduce two new specific functors, called *modulators*, L and K as follows.

Definition 7. Let $\mathbf{C}^{\mathbf{J}}$ be a category of functors from some small index category $\mathbf{J}$ with indexed objects a_j, $1 \le j \le N$ and indexed arrows $l_{j,m} : a_j \to a_m$, into the category $\mathbf{C}$. Then for each n-dimensional level $\mathbf{C}_n$, $n \ge 1$, with $\mathbf{C}_{n+1} = \mathbf{C}_n \downarrow \mathbf{C}_n$, there the following two functors: $L = (L^0, L^1) : (\mathbf{C}_n^{\mathbf{J}} \downarrow \mathbf{C}_n^{\mathbf{J}}) \to (\mathbf{C}_n \downarrow \mathbf{C}_n)^{\mathbf{J}}$ and $K : (\mathbf{C}_n \downarrow \mathbf{C}_n)^{\mathbf{J}} \to (\mathbf{C}_n^{\mathbf{J}} \downarrow \mathbf{C}_n^{\mathbf{J}})$ are defined by:

1. The object component L^0 of the functor $L : (\mathbf{C}_n^{\mathbf{J}} \downarrow \mathbf{C}_n^{\mathbf{J}}) \to (\mathbf{C}_n \downarrow \mathbf{C}_n)^{\mathbf{J}}$ is defined by:
   For every object $J(h) \in \mathrm{Ob}_{(\mathbf{C}_n^{\mathbf{J}} \downarrow \mathbf{C}_n^{\mathbf{J}})}$, where $h : G \xrightarrow{\cdot} H$ is a natural transformation between functors $G, H : \mathbf{J} \to \mathbf{C}_n$, we obtain the functor $F = L^0(J(h)) : \mathbf{J} \to (\mathbf{C}_n \downarrow \mathbf{C}_n)$, such that for each indexed object a_j in $\mathbf{J}$,

$$F(a_j) \triangleq J(h(a_j)) \tag{2.7}$$

where $h(a_i) : G(a_i) \to H(a_i)$ is the i-th arrow component of the natural transformation h. For each arrow $(g_1; g_2) : J(h) \to J(k)$ in $(\mathbf{C}_n^{\mathbf{J}} \downarrow \mathbf{C}_n^{\mathbf{J}})$, which represents the commutative diagram of natural transformation in $\mathbf{C}_n^{\mathbf{J}}$, composed by vertical composition $g_2 \bullet h = k \bullet g_1$, such that for each indexed object a_j in $\mathbf{J}$, we have the commutative diagram in $\mathbf{C}_n$, equivalent to vertical arrow in $(\mathbf{C}_n \downarrow \mathbf{C}_n)$,

$$\begin{array}{ccc} G(a_j) & \xrightarrow{h(a_j)} & H(a_j) \\ {\scriptstyle g_1(a_j)}\downarrow & & \downarrow{\scriptstyle g_2(a_j)} \\ G'(a_j) & \xrightarrow{k(a_j)} & H'(a_j) \end{array} \quad \Leftrightarrow \quad \begin{array}{c} F(a_j) = J(h(a_j)) \\ \alpha(a_j) = \downarrow (g_1(a_j); g_2(a_j)) \\ F'(a_j) = J(k(a_j)) \end{array} \tag{2.8}$$

So, the arrow component L^1 is defined by a natural transformation $L^1(g_1; g_2) = \alpha : F \xrightarrow{\cdot} F'$ between the functors $F = L^0(J(h))$ and $F' = L^0(J(k))$, such that for each indexed object a_j in $\mathbf{J}$,

$$\alpha(a_j) \triangleq (g_1(a_j); g_2(a_j)), \tag{2.9}$$

is the vertical arrow on the right-hand side of scheme (2.8).

2. The object component K^0 of the functor $K : (\mathbf{C}_n \downarrow \mathbf{C}_n)^{\mathbf{J}} \to (\mathbf{C}_n^{\mathbf{J}} \downarrow \mathbf{C}_n^{\mathbf{J}})$ is defined by: For every object (functor) F in $(\mathbf{C}_n \downarrow \mathbf{C}_n)^{\mathbf{J}}$, we define $K^0(F) = J(h)$ where h is an arrow in $\mathbf{C}_n^{\mathbf{J}}$, and hence a natural transformation between some functors $G, H : \mathbf{J} \to \mathbf{C}_n$, that must satisfy for each a_j-indexed arrow component,

$$h(a_j) \triangleq \psi(F(a_j)), \tag{2.10}$$

i. e., $J(h(a_j)) = F(a_j)$. Let, analogously, $K^0(F') = J(k)$ for an arrow (natural transformation) k in $\mathbf{C}_n^{\mathbf{J}}$.
For each arrow $\alpha : F \xrightarrow{\cdot} F'$ in $(\mathbf{C}_n \downarrow \mathbf{C}_n)^{\mathbf{J}}$, which is a natural transformation, the arrow $K^1(\alpha) : K^0(F) \to K^0(F')$ is an arrow in the arrow category $(\mathbf{C}_n \downarrow \mathbf{C}_n)^{\mathbf{J}}$, and hence represented by a pair of arrows $(g_1, g_2) = K^1(\alpha)$, i. e., $K^1(\alpha) : J(h) \to J(k)$, so that this arrow represents a commutative diagram of vertical composition of natural transformations $g_2 \bullet h = k \bullet g_1$. So, natural transformation $K^1(\alpha)$ is defined in the way that for each j-th object a_j in $\mathbf{J}$,

$$(K^1(\alpha))(a_j) \triangleq (g_1(a_j); g_2(a_j)). \tag{2.11}$$

It is easy to verify that L and K are well-defined, i. e., they map identity arrows (g_1, g_2) (when $h = k$ and $g_1 = \mathrm{dom}(h)$, $g_2 = \mathrm{cod}(h)$) into identity arrows. The functorial property for the composition of arrows is satisfied by the commutativity of the

extension of commutative diagrams in (2.8) for each indexed object a_j in $\mathbf{J}$. That is, for a composition of two arrows $l \circ g$ in $(\mathbf{C}_n^{\mathbf{J}} \downarrow \mathbf{C}_n^{\mathbf{J}})$, with $g = (g_1; g_2)$ and $l = (l_1; l_2)$, let us show that $L(l \circ g) = L(l) \bullet L(g)$ is just the vertical composition of the natural transformations $L(l)$ and $L(g)$, by using their components for each object a_j in $\mathbf{J}$:

$$\begin{aligned}
L(l \circ g)(a_j) &= L((l_1; l_2) \circ (g_1; g_2))(a_j) \\
&= L(l_1 \bullet g_1; l_2 \bullet g_2)(a_j) \quad \text{because } l_1, l_2, g_1 \text{ and } g_2 \text{ are nat. transformations} \\
&= ((l_1 \bullet g_1)(a_j); (l_2 \bullet g_2)(a_j)) \quad \text{from definition of } L \\
&= (l_1(a_j) \circ g_1(a_j); l_2(a_j) \circ g_2(a_j)) \\
&\qquad \text{from definition of vertical composition of nat. transf.} \\
&= (l_1(a_j); l_2(a_j)) \circ (g_1(a_j); g_2(a_j)) \\
&= L(l_1; l_2)(a_j) \circ L(g_1; g_2)(a_j).
\end{aligned}$$

Let us show that K is right adjoint of L, and that they are isomorphisms.

Proposition 3. *For each n-dimensional level $\mathbf{C}_n$, $n \geq 1$, there the following isomorphism of the categories $L = (L^0, L^1) : (\mathbf{C}_n^{\mathbf{J}} \downarrow \mathbf{C}_n^{\mathbf{J}}) \to (\mathbf{C}_n \downarrow \mathbf{C}_n)^{\mathbf{J}}$, and its right adjoint functor $K = (K^0, K^1) : (\mathbf{C}_n \downarrow \mathbf{C}_n)^{\mathbf{J}} \to (\mathbf{C}_n^{\mathbf{J}} \downarrow \mathbf{C}_n^{\mathbf{J}})$, with*

$$L \circ K = I_{(\mathbf{C}_n \downarrow \mathbf{C}_n)^{\mathbf{J}}} \quad \textit{and} \quad K \circ L = I_{(\mathbf{C}_n^{\mathbf{J}} \downarrow \mathbf{C}_n^{\mathbf{J}})} \tag{2.12}$$

where $I_{(\mathbf{C}_n \downarrow \mathbf{C})_n^{\mathbf{J}}}$ and $I_{(\mathbf{C}_n^{\mathbf{J}} \downarrow \mathbf{C}_n^{\mathbf{J}})}$ are two identity functors, relatively. This adjunction is denoted by $(L, K, \mathrm{id}_{LK}, \mathrm{id}_{KL})$ where the identity natural transformation $\mathrm{id}_{KL} : I_{(\mathbf{C}_n^{\mathbf{J}} \downarrow \mathbf{C}_n^{\mathbf{J}})} \to KL$ is the unit and $\mathrm{id}_{LK} : LK \to I_{(\mathbf{C}_n \downarrow \mathbf{C}_n)^{\mathbf{J}}}$ is the counit of this adjunction.

In the case when $\mathbf{J} = \mathbf{2}$ is the category of only two indexed objects a_1 and a_2 with the unique nonidentity arrow $l_{12} : a_1 \to a_2$, we obtain that

$$L = K = L_{\perp} : \mathbf{C}_{n+2} \to \mathbf{C}_{n+2},$$

i. e., K and L are equal (up to isomorphism) to the duality endofunctor $L_{\perp}$ in (1.10) *(the objects $\mathbf{C}_n^{\mathbf{2}}$ and $\mathbf{C}_{n+1} = (\mathbf{C}_n \downarrow \mathbf{C}_n)$ are equal up to the isomorphisms in the **Cat** category).*

Proof. The equations in (2.12) are easy to verify directly from definitions of the functors L and K. In order to show that, in the case when $\mathbf{J} = \mathbf{2}$, we obtain that $\mathbf{C}_n^{\mathbf{2}}$ is equal (up to isomorphism) to $(\mathbf{C}_n \downarrow \mathbf{C}_n)$ in **Cat**, and we define the functor $I_s : \mathbf{C}_n^{\mathbf{2}} \to (\mathbf{C}_n \downarrow \mathbf{C}_n)$ by: for each object $F : \mathbf{2} \to \mathbf{C}_n$, we set $I_s(F) \triangleq \langle F(a_1), F(a_2), F(e) \rangle = J(F(e))$, and arrow (nat. transf.) $\eta : F \dashrightarrow F'$ in $\mathbf{C}_n^{\mathbf{2}}$, we set $I_s(\eta) \triangleq (\eta(a_1); \eta(a_2)) : J(F(e)) \to J(F'(e))$.

We define the inverse functor $I_s^{-1} : (\mathbf{C}_n \downarrow \mathbf{C}_n) \to \mathbf{C}_n^{\mathbf{2}}$ by: for each object $J(h)$ in $\mathbf{C}_{n+1} = (\mathbf{C}_n \downarrow \mathbf{C}_n)$, we set $F \triangleq I_s^{-1}(J(h)) : \mathbf{2} \to \mathbf{C}_n$ is the functor such that $F(a_1) = \mathrm{dom}(h)$ and $F(a_2) = \mathrm{cod}(h)$. While, for the arrow $(g_1; g_2) : J(h) \to J(k)$, we set that $\eta \triangleq I_s^{-1}(g_1; g_2)$ is the natural transformation from the functor $F = I_s^{-1}(J(h))$ into the functor $F' = I_s^{-1}(J(k))$,

defined by its arrow components, $\eta(a_1) = g_1$ and $\eta(a_2) = g_2$. So, their compositions, $I_s^{-1} I_s = \mathrm{Id}_{\mathbf{C}_n^2}$ and $I_s I_s^{-1} = \mathrm{Id}_{\mathbf{C}_n \downarrow \mathbf{C}_n}$, are the identity endofunctors.

Thus, with this isomorphism in mind,[1] we can write that $\mathbf{C}_n^{\mathbf{2}} = (\mathbf{C}_n \downarrow \mathbf{C}_n)$, and hence $(\mathbf{C}_n^{\mathbf{J}} \downarrow \mathbf{C}_n^{\mathbf{J}}) = (\mathbf{C}_n \downarrow \mathbf{C}_n)^{\mathbf{J}} = \mathbf{C}_{n+2}$, i. e., both L and K in this particular case when $\mathbf{J} = \mathbf{2}$ become two endofunctors of $\mathbf{C}_{n+2}$ and equivalent to the duality endofunctor $L_\perp$. □

2.2 Comma-propagation of functors and natural transformations

So, as in the case of the comma-induction for the covariant functors in Definition 3, the comma-propagation for the functors is the following generalization of the comma-induction operations [4], denoted by $\{_\}_n$, for $n \geq 1$, of the functors. Here, however, we will simplify this denotation by using *a unique comma-propagation operation* $\{_\}$ and by denoting the comma-propagation derived functors by F_n and the comma-propagated natural transformations by τ_n so that we obtain coherent denotation with the comma-induction operation. That is important also because we will show that the comma-induction is only a particular case of more general comma-propagation (the case when domain or codomain category of a given base functor F can be a category of functors as well).

Definition 8. For a given small index category $\mathbf{J}$, we define the following four cases of the comma-propagation of a given base covariant functor F (considered as 1-level), which for each n-dimensional level, will be denoted by the F_n comma-propagated functor with the n-dimensional levels $\mathbf{C}_n$ and $\mathbf{D}_n$, as follows for $n \geq 2$:

1. Basic case: when the base functor is $F : \mathbf{D} \to \mathbf{C}$, with $F_1 \triangleq F$, and let $F_{n-1} : \mathbf{D}_{n-1} \to \mathbf{C}_{n-1}$ be the $(n-1)$-dimensional comma-propagated functor. Then the n-dimensional comma-propagated functor is its comma-lifted

$$F_n = \{F_{n-1}\} \triangleq \widehat{F}_{n-1} : \mathbf{D}_n \to \mathbf{C}_n,$$

   i. e., equal to the comma-induced functor F_n in Definition 3.
2. Descending case: when the base functor is $F : \mathbf{D}^{\mathbf{J}} \to \mathbf{C}$, with $F_1 \triangleq F$, and let $F_{n-1} : \mathbf{D}_{n-1}^{\mathbf{J}} \to \mathbf{C}_{n-1}$ be the $(n-1)$-dimensional comma-propagated functor. Then, for $K : \mathbf{D}_n^{\mathbf{J}} \to (\mathbf{D}_{n-1}^{\mathbf{J}} \downarrow \mathbf{D}_{n-1}^{\mathbf{J}})$, the n-dimensional comma-propagated functor is

$$F_n = \{F_{n-1}\} \triangleq \widehat{F}_{n-1} \circ K : \mathbf{D}_n^{\mathbf{J}} \to \mathbf{C}_n.$$

---

1 But this is now an identity between objects, and this is simply not given a priori in our categorial language. We are thus lead, for purely conceptual reasons, to the notion of equivalence of categories. In that definition, we simply mentioned natural isomorphisms between functors.

3. Ascending case: when the base functor is $F : \mathbf{D} \to \mathbf{C}^{\mathbf{J}}$, with $F_1 \triangleq F$, and let $F_{n-1} : \mathbf{D}_{n-1} \to \mathbf{C}^{\mathbf{J}}_{n-1}$ be the $(n-1)$-dimensional comma-propagated functor. Then, for $L : (\mathbf{C}^{\mathbf{J}}_{n-1} \downarrow \mathbf{C}^{\mathbf{J}}_{n-1}) \to \mathbf{C}^{\mathbf{J}}_{n}$, the n-dimensional comma-propagated functor is

$$F_n = \{F_{n-1}\} \triangleq L \circ \widehat{F}_{n-1} : \mathbf{D}_n \to \mathbf{C}^{\mathbf{J}}_{n}.$$

4. Balanced case: when the base functor is $F : (\mathbf{D} \downarrow \mathbf{D}) \to (\mathbf{C} \downarrow \mathbf{C})$, with $F_1 \triangleq F$, and let $F_{n-1} : \mathbf{D}^{\mathbf{J}}_{n-1} \to \mathbf{C}^{\mathfrak{J}}_{n-1}$ be the $(n-1)$-dimensional comma-propagated functor. Then, for $K : \mathbf{D}^{\mathbf{J}}_{n} \to (\mathbf{D}^{\mathbf{J}}_{n-1} \downarrow \mathbf{D}^{\mathbf{J}}_{n-1})$ and $L : (\mathbf{C}^{\mathfrak{J}}_{n-1} \downarrow \mathbf{C}^{\mathfrak{J}}_{n-1}) \to \mathbf{C}^{\mathfrak{J}}_{n}$, the n-dimensional comma-propagated functor is

$$F_n = \{F_{n-1}\} \triangleq L \circ \widehat{F}_{n-1} \circ K : \mathbf{D}^{\mathbf{J}}_{n} \to \mathbf{C}^{\mathfrak{J}}_{n}$$

where $\mathfrak{J}$ is another small index category.

The functors K and L in Definition 7 are initial/final *modulators* relatively: K is used to pass from the category of functors into an arrow category, from the fact that the comma-lifted functor works only for the hierarchy of arrow categories, while L is used to pass from an arrow category (the result of comma lifting) into final category of functors.

Consequently, in the basic case, the comma-propagated functor F_n is just the n-times iteration of the comma lifting of a functor defined by (1.7), $\widehat{G} \triangleq (JG^1\psi, (G^1F^1_{\mathrm{st}}; G^1S^1_{\mathrm{nd}}))$, exactly as we obtained the comma-induced functor F_n. For the remaining three cases, modulated by the functors K and L in Definition 7, the comma-propagation reduces to the comma-induction in the particular cases when the small index categories $\mathbf{J}$ and $\mathfrak{J}$ are equal to the one-arrow index category $\mathbf{2}$.

Let us show in an example how the comma-propagation of functors preserves their original properties in the higher n-dimensional levels.

Example 4. Let us consider the general diagonal functor (which is an ascending case) $\triangle : \mathbf{D} \to \mathbf{C}^{\mathbf{J}}$ in Definition 6, when $\mathbf{C} = \mathbf{D}$ with $m = 1$, i. e., $\triangle : \mathbf{C} \to \mathbf{C}^{\mathbf{J}}$, and suppose that for $n \geq 1$ for the comma-propagated diagonal functor $\triangle_n : \mathbf{C}_n \to \mathbf{C}^{\mathbf{J}}_{n}$ it holds the diagonal property, with $\triangle_n = \triangle$. That is, for each object $c \in \mathrm{Ob}_{\mathbf{C}_n}$, the object (a functor) $\triangle^0_n(c) : \mathbf{J} \to \mathbf{C}_n$ satisfies the diagonal property in point 1 of Definition 6, such that for all indexed objects a_i in $\mathbf{J}$,

$$\triangle^0_n(c)(a_j) = c$$

and that for each arrow $g \in \mathrm{Mor}_{\mathbf{C}_n}$, the arrow (a natural transformation) $\triangle^1_n(g)$ satisfies the diagonal property 2 of Definition 6, such that for all indexed objects a_i in $\mathbf{J}$,

$$\triangle^1_n(g)(a_j) = g.$$

Let us show that these diagonal properties are valid also for the comma-propagated functor $\triangle_{n+1} : \mathbf{C}_{n+1} \to \mathbf{C}^{\mathbf{J}}_{n+1}$, and consider an arrow $(g_1; g_2) : J(k_1) \to J(k_2)$ in $\mathbf{C}_{n+1} = (\mathbf{C}_n \downarrow \mathbf{C}_n)$, for which there exists the commutative diagram $g_2 \circ k_1 = k_2 \circ g_1$ in the lower n-dimensional level $\mathbf{C}_n$. From the fact that the base functor $\triangle : \mathbf{C} \to \mathbf{C}^{\mathbf{J}}$ is an ascending case, we obtain the object (functor):

$$\begin{aligned} F = \triangle_{n+1}(J(k_1)) &= L \circ \widehat{\triangle}_n(J(k_1)) \\ &= L^0(J\triangle^1_n\psi)(J(k_1)) \\ &= L^0 J(\triangle^1_n(k_1)) \end{aligned}$$

where $h = \triangle^1_n(k_1)$ is an arrow (a natural transformation) in $\mathbf{C}^{\mathbf{J}}_n$. Thus, from (2.7), for each indexed object a_j in $\mathbf{J}$,

$$\begin{aligned} \triangle_{n+1}(J(k_1))(a_j) = F(a_j) = J(h(a_j)) &= J(\triangle^1_n(k_1)(a_j)) \\ &= J(k_1), \quad \text{by inductive hypothesis and (2.4)}, \end{aligned}$$

so that the object-component of the comma-propagated functor $\triangle_{n+1}$ satisfies the diagonal property. Let us show that the arrow-component of the functor $\triangle_{n+1}$ satisfies the diagonal property, i. e., for each arrow $(g_1; g_2)$ in $\mathbf{C}_{n+1}$, the arrow $\eta = \triangle_{n+1}(g_1; g_2)$ in $\mathbf{C}^{\mathbf{J}}_{n+1}$ is a natural transformation, and hence for each index object a_j in $\mathbf{J}$, we have that

$$\begin{aligned} (\triangle_{n+1}(g_1; g_2))(a_j) &= (L \circ \widehat{\triangle}_n(g_1; g_2))(a_j) \\ &= (L^1(\triangle^1_n F^1_{\mathrm{st}}; \triangle^1_n S^1_{\mathrm{nd}})(g_1; g_2))(a_j) \\ &= (L^1(\triangle^1_n(g_1); \triangle^1_n(g_2)))(a_j) \quad \text{for nat. transf. } \triangle^1_n(g_i),\ i = 1, 2 \text{ (arrows in } \mathbf{C}^{\mathbf{J}}_n) \\ &= (\triangle^1_n(g_1)(a_j); \triangle^1_n(g_2)(a_j)) \quad \text{from (2.9)} \\ &= (g_1; g_2), \quad \text{from inductive hypothesis}, \end{aligned}$$

so that the arrow-component of the comma-propagated functor $\triangle_{n+1}$ satisfies the diagonal property as well.

The comma-propagation can be applied also to natural transformations, as in the case of the comma-induction in Definition 4.

Definition 9. We define the following four cases of the comma-propagation of a given base natural transformation $\tau : F \xrightarrow{\cdot} G$ (considered as 1-dimensional level denoted by τ_1) which for each $(n-1)$-dimensional level, $n \geq 2$, will be denoted by $\tau_{n-1} : F_{n-1} \xrightarrow{\cdot} G_{n-1}$ comma-propagated natural transformation, as follows:

1. Basic case: when the base functors are $F, G : \mathbf{D} \to \mathbf{C}$. Then n-dimensional comma-propagated natural transformation is comma lifted (from (1.8)),

$$\tau_n = \{\tau_{n-1}\} \triangleq \widehat{\tau}_{n-1} : \widehat{F}_{n-1} \xrightarrow{\cdot} \widehat{G}_{n-1},$$

i. e., the function $\tau_n = (\tau_{n-1}F^0_{\mathrm{st}}; \tau_{n-1}S^0_{\mathrm{nd}})$, as in the comma-induction.

2. Descending case: when the base functors are $F, G : \mathbf{D}^{\mathbf{J}} \to \mathbf{C}$. Then the n-dimensional comma-propagated natural transformation, for the functors in point 2 of Definition 8, is (from (A.9) in the Appendix),

$$\tau_n = \{\tau_{n-1}\} \triangleq \widehat{\tau}_{n-1}K : \widehat{F}_{n-1} \circ K \xrightarrow{\cdot} \widehat{G}_{n-1} \circ K,$$

i. e., the function $\tau_n = (\tau_{n-1}F^0_{\mathrm{st}}; \tau_{n-1}S^0_{\mathrm{nd}})K^0$.

3. Ascending case: when the base functor is $F, G : \mathbf{D} \to \mathbf{C}^{\mathbf{J}}$. Then the n-dimensional comma-propagated natural transformation, for the functors in point 3 of Definition 8, is (from (A.10) in the Appendix),

$$\tau_n = \{\tau_{n-1}\} \triangleq L\widehat{\tau}_{n-1} : L \circ \widehat{F}_{n-1} \xrightarrow{\cdot} L \circ \widehat{G}_{n-1},$$

i. e., the function $\tau_n = L^1(\tau_{n-1}F^0_{\mathrm{st}}; \tau_{n-1}S^0_{\mathrm{nd}})$.

4. Balanced case: when the base functors are $F, G : \mathbf{D}^{\mathbf{J}} \to \mathbf{C}^{\mathfrak{J}}$, then the n-dimensional comma-propagated natural transformation, for the functors in point 4 of Definition 8 is (from (A.9) and (A.10) in the Appendix),

$$\tau_n = \{\tau_{n-1}\} \triangleq L \circ \widehat{\tau}_{n-1}K : L \circ \widehat{F}_{n-1} \circ K \xrightarrow{\cdot} L \circ \widehat{G}_{n-1} \circ K,$$

i. e., the function $\tau_n = L^1(\tau_{n-1}F^0_{\mathrm{st}}; \tau_{n-1}S^0_{\mathrm{nd}})K^0$.

So, as in the case of the comma-propagated functors, also for the comma-propagated natural transformations different from the basic case, we have the modulation by the functors K and L. It is easy to verify that in all four cases above, from Definition 3, we obtain that for each $n \geq 1$, the comma-propagated natural transformation is just between comma-propagated functors derived from the base functors F and G:

$$\tau_n : F_n \xrightarrow{\cdot} G_n.$$

The following properties for the combination of comma-propagated functors and natural transformations are shown in [4].

Proposition 4. *The following are distributive laws between the operation of comma-propagation '{_}' and the following categorial operations, for $n \geq 1$:*

1. *The operation of composition of functors:*

$$F_n \circ G_n = (F \circ G)_n.$$

2. *The horizontal composition of natural transformations:*

$$\tau_n \circ \eta_n = (\tau \circ \eta)_n.$$

3. *The vertical composition of natural transformations:*

$$\tau_n \bullet \eta_n = (\tau \bullet \eta)_n.$$

Proof. It is easy to verify by using the inductive proof, for each possible case of functors and natural transformation. For example, let us consider, for the base categories **C**, **D** and **E**, the composition of functors $F \circ G$, of a descending functor $F : \mathbf{D}^{\mathbf{J}} \to \mathbf{C}$ and an ascending functor $G : \mathbf{E} \to \mathbf{D}^{\mathbf{J}}$, and suppose that for a given $k \geq 1$ it holds that $F_k \circ G_k = (F \circ G)_k$. Let us show that it holds for $n = k + 1$ as well, i. e.,

$$\begin{aligned}
F_{k+1} \circ G_{k+1} &= (\widehat{F}_k \circ K) \circ (L \circ \widehat{G}_k) \\
&= \widehat{F_k} \circ \widehat{G_k} \quad \text{from } K \circ L \text{ an identity functor} \\
&= \left(JF_k\psi, (F_k^1 F_{\mathrm{st}}^1; F_k^1 S_{\mathrm{nd}}^1)\right) \circ \left(JG_k\psi, (G_k^1 F_{\mathrm{st}}^1; G_k^1 S_{\mathrm{nd}}^1)\right) \\
&= \left(JF_k\psi JG_k\psi, (F_k^1 F_{\mathrm{st}}^1; F_k^1 S_{\mathrm{nd}}^1)(G_k^1 F_{\mathrm{st}}^1; G_k^1 S_{\mathrm{nd}}^1)\right) \\
&= \left(JF_k G_k\psi, (F_k^1 G_k^1 F_{\mathrm{st}}^1; F_k^1 G_k^1 S_{\mathrm{nd}}^1)\right) \\
&= \widehat{F_k \circ G_k} \\
&= \widehat{(F \circ G)_k} \quad \text{from inductive hypothesis} \\
&= (F \circ G)_{k+1} \quad \text{from the fact that composed functor } F \circ G \text{ is a basic case.} \quad \square
\end{aligned}$$

There exists the following closure property for the comma-propagation, which generalizes the closure for the comma-induction in Corollary 2.

Corollary 3. *The following property of closure for the comma-propagation is valid:*

1. *Let F be a base covariant functor in Definition 3. Any functor induced from a comma-propagated functor F_n, for $n \geq 1$, is a comma-propagated functor from F (with $F_1 = F$). We denote by $\mathfrak{R}_P(F_n) \triangleq \{F_m \mid m \geq n\}$ the set of all comma-propagated functors from the base functor F_n. Then $\mathfrak{R}_P$ is an operation such that for any $n \geq 1$,*

$$\mathfrak{R}_P(F) \supseteq \mathfrak{R}_P(F_n) = \mathfrak{R}_P(F_{n+1}) \bigcup \{F_n\}.$$

2. *Let τ be a natural transformation. Any natural transformation induced from a comma-propagated natural transformation τ_n, for $n \geq 1$, is a comma-propagated natural transformation from τ (with $\tau_1 = \tau$).*
We denote by $\mathfrak{R}_{\mathrm{PT}}(\tau_n) \triangleq \{\tau_m \mid m \geq n\}$ the set of all comma-propagated natural transformations from τ_n. Then $\mathfrak{R}_{\mathrm{PT}}$ is an operation such that for any $n \geq 1$,

$$\mathfrak{R}_{\mathrm{PT}}(\tau) \supseteq \mathfrak{R}_{\mathrm{PT}}(\tau_n) = \mathfrak{R}_{\mathrm{PT}}(\tau_{n+1}) \bigcup \{\tau_n\}.$$

Proof. By induction, for each of the four possible cases:

1.1 Basic case $F : \mathbf{D} \to \mathbf{C}$. Suppose that it holds for $n = k$ and let us show it for $k + 1$, for the basic functor $G = F_{k+1} : \mathbf{D}_{k+1} \to \mathbf{C}_{k+1}$: For $m = 0$, it holds that $G_0 = G = F_{k+1}$;

suppose that for $m \geq 1$ it holds that $G_m = F_{m+k+1}$, and let us show that it is valid for $m+1$ as well:

$$\begin{aligned} G_{m+1} &= \widehat{G}_m \\ &= (JG_m\psi, (G_m^1 F_{\mathrm{st}}^1; G_m^1 S_{\mathrm{nd}}^1)) \quad \text{from definition of comma lifting} \\ &= (JF_{m+k+1}\psi, (F_{m+k+1}^1 F_{\mathrm{st}}^1; F_{m+k+1}^1 S_{\mathrm{nd}}^1)) \quad \text{from the hypothesis} \\ &= \widehat{F}_{m+k+1} \quad \text{from definition of comma lifting} \\ &= F_{m+k+2} \quad \text{by definition of the basic case for functor } F. \end{aligned}$$

1.2 Descending case $F : \mathbf{D}^{\mathbf{J}} \to \mathbf{C}$. Suppose that it holds for $n = k$ and let us show it for $k+1$, for the basic functor $G = F_{k+1} : \mathbf{D}_{k+1}^{\mathbf{J}} \to \mathbf{C}_{k+1}$: For $m = 0$, it holds that $G_0 = G = F_{k+1}$; suppose that for $m \geq 1$ it holds that $G_m = F_{m+k+1}$, and let us show that it is valid for $m+1$ as well:

$$\begin{aligned} G_{m+1} &= \widehat{G}_m \\ &= (JG_m\psi K^0, (G_m^1 F_{\mathrm{st}}^1; G_m^1 S_{\mathrm{nd}}^1)K^1) \quad \text{from the definition of comma lifting} \\ &= (JF_{m+k+1}\psi K^0, (F_{m+k+1}^1 F_{\mathrm{st}}^1; F_{m+k+1}^1 S_{\mathrm{nd}}^1)K^1) \quad \text{from the hypothesis} \\ &= \widehat{F}_{m+k+1} \circ K \quad \text{from the definition of comma lifting} \\ &= F_{m+k+2} \quad \text{by the definition of the descending case for the functor } F. \end{aligned}$$

1.3 Ascending case $F : \mathbf{D} \to \mathbf{C}^{\mathbf{J}}$. Suppose that it holds for $n = k$ and let us show it for $k+1$, for the basic functor $G = F_{k+1} : \mathbf{D}_{k+1} \to \mathbf{C}_{k+1}^{\mathbf{J}}$: For $m = 0$, it holds that $G_0 = G = F_{k+1}$; suppose that for $m \geq 1$ it holds that $G_m = F_{m+k+1}$, and let us show that it is valid for $m+1$ as well:

$$\begin{aligned} G_{m+1} &= \widehat{G}_m \\ &= (L^0 JG_m\psi, L^1(G_m^1 F_{\mathrm{st}}^1; G_m^1 S_{\mathrm{nd}}^1)) \quad \text{from the definition of comma lifting} \\ &= (L^0 JF_{m+k+1}\psi, L^1(F_{m+k+1}^1 F_{\mathrm{st}}^1; F_{m+k+1}^1 S_{\mathrm{nd}}^1)) \quad \text{from the hypothesis} \\ &= L \circ \widehat{F}_{m+k+1} \quad \text{from the definition of comma lifting} \\ &= F_{m+k+2} \quad \text{by the definition of the basic case for the functor } F. \end{aligned}$$

1.4 Balanced case $F : \mathbf{D}^{\mathbf{J}} \to \mathbf{C}^{\mathfrak{J}}$. Combination of both cases 1.2 and 1.3 above.

Analogously, we can show for the natural transformations $\tau : F \to G$:

2.1 Basic case, with functors $F, G : \mathbf{D} \to \mathbf{C}$. Suppose that it holds for $n = k$ and let us show it for $k+1$, for the basic natural transformation $\sigma = \tau_{k+1} : F_{k+1} \xrightarrow{\cdot} G_{k+1}$: For $m = 0$, it holds that $\sigma_0 = \sigma = \tau_{k+1}$; suppose that for $m \geq 1$ it holds that $\sigma_m = \tau_{m+k+1}$, and let us show that it is valid for $m+1$ as well:

$$\begin{aligned}\sigma_{m+1} &= \widehat{\sigma}_m \\ &= (\sigma^1_m F^0_{\mathrm{st}}; \sigma^1_m S^0_{\mathrm{nd}}) \quad \text{from the definition of comma lifting} \\ &= (\tau^1_{m+k+1} F^0_{\mathrm{st}}; \tau^1_{m+k+1} S^0_{\mathrm{nd}}) \quad \text{from the hypothesis} \\ &= \widehat{\tau}_{m+k+1} \quad \text{from the definition of comma lifting} \\ &= \tau_{m+k+2} \quad \text{by the definition of the basic case for } \tau.\end{aligned}$$

2.2 Descending case, with functors $F, G : \mathbf{D}^{\mathbf{J}} \to \mathbf{C}$. Suppose that it holds for $n = k$ and let us show it for $k+1$, for the basic natural transformation $\sigma = \tau_{k+1} : F_{k+1} \xrightarrow{\cdot} G_{k+1}$: For $m = 0$, it holds that $\sigma_0 = \sigma = \tau_{k+1}$; suppose that for $m \geq 1$ it holds that $\sigma_m = \tau_{m+k+1}$, and let us show that is valid for $m+1$ as well:

$$\begin{aligned}\sigma_{m+1} &= \widehat{\sigma}_m \\ &= (\sigma^1_m F^0_{\mathrm{st}}; \sigma^1_m S^0_{\mathrm{nd}})K \quad \text{from the definition of comma lifting} \\ &= (\tau^1_{m+k+1} F^0_{\mathrm{st}}; \tau^1_{m+k+1} S^0_{\mathrm{nd}})K \quad \text{from the hypothesis} \\ &= \widehat{\tau}_{m+k+1} K \quad \text{from the definition of comma lifting} \\ &= \tau_{m+k+2} \quad \text{by the definition of the descending case for } \tau.\end{aligned}$$

2.3 The ascending case is analogous to the point 2.2, by using modulated functors L as in the point 1.3.
The balanced case is combination of cases 2.2 and 2.3. □

2.3 Comma-propagation of (co)limits

Adjunctions, $(F, G, \varepsilon, \eta)$ of the functor $F : \mathbf{C} \to \mathbf{D}$ with its right-adjoint functor $G : \mathbf{D} \to \mathbf{C}$, which generate two natural transformations (the unit $\eta : I_{\mathbf{C}} \xrightarrow{\cdot} GF$ and counit $\varepsilon : FG \xrightarrow{\cdot} I_{\mathbf{D}}$, which in the dual version of the adjunction have opposite arrow directions) introduced by Kan in 1958, provide a descriptive framework of great generality, capturing the essence of many canonical constructions. They turn up throughout mathematics often as “closures” and “completions” and as “free” and “generated” structures. For example, the transitive closure of a graph, the completion of a metric space, factor commutator groups and free algebras are all examples of adjunctions. In categorical logic, quantifiers are interpreted as adjunctions with respect to the substitution of variables.

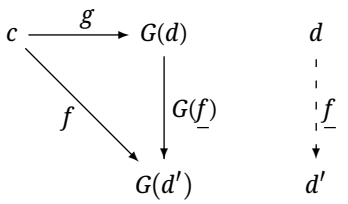
The canonical nature of these constructions is captured by universality. Indeed, adjunctions subsume the universal structures, both limits and colimits, when $\mathbf{D} = \mathbf{C}^{\mathbf{J}}$ is a category of functors for a small indexed category $\mathbf{J}$ and $F = \triangle$ is the diagonal functor. Fundamentally, the adjunctions represent two “forces” generated by the composition of these two functors creating the unit natural transformation such that for each object c in the category $\mathbf{C}$ is generated the displacement (arrow) toward the object $GF(c)$, and

by the counit natural transformation such that for each object d in the category $\mathbf{D}$, is generated the displacement (arrow) from the object $FG(d)$ into this object d. From the fact that these forces are applied to *all* objects of these two categories, we can consider an adjunction as a kind of the scalar *fields* in these two categories: *adjunctions-as-fields*. More about this interpretation of the adjunction will be discussed in Section 2.5.

(Co)limits are a generalization of the universally defined structures. The generalization is based upon diagrams in a category. We define diagrams and then the general structure of a (co)limit. Limits and colimits can both be defined through universality. More examples will be found in next section. A curious result is the interdefinability of universal concepts in category theory. Universal concepts can be obtained from one another by interpreting them in suitable categories. Limits and colimits give a unified treatment of constructs such as products and sums of pairs of objects and limits of chains of objects.

Both of these important categorial concepts are built by using the more simple concepts of universal arrows. By considering that, from Theorem 1, the comma lifting preserves the universal arrows, and the next theorem demonstrates the universal arrows are preserved by the comma-propagation.

Theorem 3. *Given a functor G, for each universal arrow* "from c to functor G," *which is the pair* (d, g)*, with object* d *in the domain category of* G *and arrow* $g : c \to G(d)$ *in the codomain category of* G*, presented by scheme* (A.5)[2] *in the Appendix,*



(2.13)

with $G_1 = G$*, there is the comma-propagated universal arrow as follows:*

1. *Basic case: when* $G : \mathbf{D} \to \mathbf{C}$*, then we have the universal arrow* $(J(\mathrm{id}_d), (g; g))$ *from* $J(\mathrm{id}_c)$ *to the comma-propagated functor* $G_2 : (\mathbf{D} \downarrow \mathbf{D}) \to (\mathbf{C} \downarrow \mathbf{C})$*.*
2. *Descending case: when* $G : \mathbf{D}^{\mathbf{J}} \to \mathbf{C}$*, then we have the universal arrow* $(L^0J(\mathrm{id}_d), (g; g))$ *from* $L^0J(\mathrm{id}_c)$ *to the comma-propagated functor* $G_2 : (\mathbf{D} \downarrow \mathbf{D})^{\mathbf{J}} \to (\mathbf{C} \downarrow \mathbf{C})$*.*
3. *Ascending case: when* $G : \mathbf{D} \to \mathbf{C}^{\mathbf{J}}$*, then we have the universal arrow* $(J(\mathrm{id}_d), L^1(g; g))$ *from* $J(\mathrm{id}_c)$ *to the comma-propagated functor* $G_2 : (\mathbf{D} \downarrow \mathbf{D}) \to (\mathbf{C} \downarrow \mathbf{C})^{\mathbf{J}}$*.*
4. *Balanced case: when* $G : \mathbf{D}^{\mathbf{J}} \to \mathbf{C}^{\mathfrak{J}}$*, then we have the universal arrow* $(L^0J(\mathrm{id}_d), L^1(g; g))$ *from* $L^0J(\mathrm{id}_c)$ *to the comma-propagated functor* $G_2 : (\mathbf{D} \downarrow \mathbf{D})^{\mathbf{J}} \to (\mathbf{C} \downarrow \mathbf{C})^{\mathfrak{J}}$*,*

where L *is a modulator functor isomorphism specified in point 1 of Definition* 7*.*

---

2 Where for each pair (d', f), there exists *the unique* arrow $\underline{f} : d \to d'$ such that $f = G(\underline{f}) \circ g$.

These results hold also for the dual concept of the couniversal arrow "from functor G to c," *represented by scheme* (A.6) *in the Appendix, with the opposite direction of all arrows in scheme* (2.13) *above.*

Proof. Let us show for each particular case, by considering that $G_1 = G$:

1. Basic case: when coma-propagated functor G_2 is equal to comma lifted functor $\widehat{G}$, so the proof is given in Theorem 1.

2. Descending case: Let us show that for each object $J(k)$ in the comma-propagated category $\{\mathbf{C}\} = \mathbf{C}_2 = \mathbf{C} \downarrow \mathbf{C}$, with $k : c_1 \to c_2$ an arrow in $\mathbf{C}$, there is the universal arrow (d, g) with a functor $d : \mathbf{J} \to (\mathbf{D} \downarrow \mathbf{D})$ and $g = (g_1; g_2)$ where g_1, g_2 are two arrows of $\mathbf{C}$, such that for each pair (d', f) with $f = (f_1; f_2)$ where f_1, f_2 are two arrows of $\mathbf{C}$, there is the unique arrow (natural transformation) $\underline{f} : d \to d'$ such that the following diagram commutes:

$$
\begin{array}{ccccc}
J(k) & \xrightarrow{(g_1;g_2)} & \widehat{G}(J(l)) & \qquad J(l) = K^0(d) & \qquad d \\
& \searrow^{f=(f_1;f_2)} & \downarrow{\scriptstyle \widehat{G}(\underline{f}_1;\underline{f}_2)} & \downarrow{\scriptstyle (\underline{f}_1;\underline{f}_2) = K^1(\underline{f})} & \vdots\,{\scriptstyle \underline{f}} \\
& & \widehat{G}(J(h)) & \qquad J(h) = K^0(d') & \qquad d'
\end{array}
\tag{2.14}
$$

$$
(\mathbf{C} \downarrow \mathbf{C}) \xleftarrow{\widehat{G}} (\mathbf{D}^{\mathbf{J}} \downarrow \mathbf{D}^{\mathbf{J}}) \xleftarrow{K} (\mathbf{D} \downarrow \mathbf{D})^{\mathbf{J}}
$$

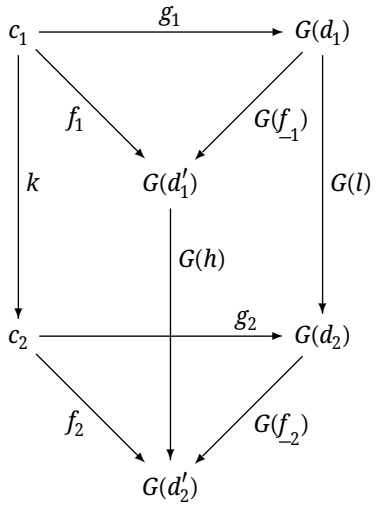
$$
(\mathbf{C} \downarrow \mathbf{C}) \xleftarrow{G_2 = \widehat{G} \circ K} (\mathbf{D} \downarrow \mathbf{D})^{\mathbf{J}}
$$

so that for each pair (d', f) where the object d' in $(\mathbf{D} \downarrow \mathbf{D})^{\mathbf{J}}$ is a functor $d' = F : \mathbf{J} \to (\mathbf{D} \downarrow \mathbf{D})$ so that, from point 2 of Definition 7, we can define $K^0(d') = K^0(F) = J(h)$ where $h : d'_1 \xrightarrow{\cdot} d'_2$ is a natural transformation between functors d'_1, d'_2 in $\mathbf{D}^{\mathbf{J}}$ that must satisfy (2.10), for $H_1 = d'_1$ and $H_2 = d'_2$, (2.10) for each indexed object a_j of the small indexed category $\mathbf{J}$, i. e., $h(a_j) = \psi(F(a_j)) = \psi(d'(a_j))$.

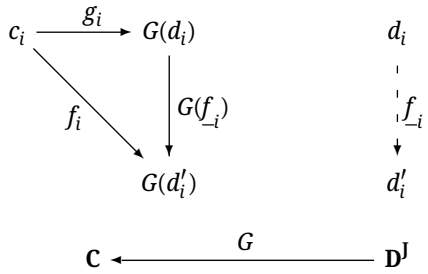
Analogously, we define $J(l) = K^0(d)$, with $l : d_1 \xrightarrow{\cdot} d_2$ is a natural transformation between functors d_1, d_2 in $\mathbf{D}^{\mathbf{J}}$, and arrow $(\underline{f}_1;\underline{f}_2) = K^1(\underline{f})$, so that this vertical arrow above is the commutative diagram in $\mathbf{D}^{\mathbf{J}}$, $h \bullet \underline{f}_1 = \underline{f}_2 \bullet l$, of vertical composition of natural transformations. We have that in the commutative diagram in $(\mathbf{C} \downarrow \mathbf{C})$,

$$
\begin{aligned}
\widehat{G}(J(l)) &= JG^1\psi(J(l)) = J(G^1(l)), \\
\widehat{G}(\underline{f}_1;\underline{f}_2) &= (G^1F^1_{\mathrm{st}}; G^1S^1_{\mathrm{nd}})(\underline{f}_1;\underline{f}_2) = (G^1(\underline{f}_1); G^1(\underline{f}_2)), \\
\widehat{G}(J(h)) &= JG^1\psi(J(h)) = J(G^1(h)),
\end{aligned}
$$

so that this diagram corresponds to the following commutative diagram in **C**:



Two commutative triangles above correspond to the following two universal arrow diagrams in **C** for $i = 1, 2$,



so that from the universal arrow (d_i, g_i) for (each) pair (d_i', f_i) the unique arrow $\underline{f}_i$ is unique for which diagram commutes. Thus, the arrow $\underline{f} = L(\underline{f}_1; \underline{f}_2)$ is unique as well, and hence the diagram (2.14) is the comma-propagated universal arrow diagram.

The other two cases, ascending and balanced, are easy to verify by using the same method used for the descending case above. □

For example, let us consider the following couniversal arrows, for the general diagonal functor in Definition 6 with $\mathbf{C} = \mathbf{D} \times \mathbf{D}$ and $\mathbf{J} = \mathbf{1}$ is the category of only one object a_1 and its identity arrow id_{a_1} with scheme (2.6).

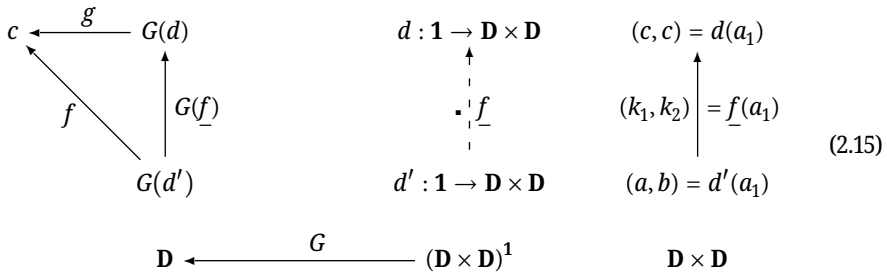
Example 5. In this case, $\triangle : \mathbf{D} \to (\mathbf{D} \times \mathbf{D})^{\mathbf{1}}$ (which is an ascending functor is discussed in detail in Example 4).

Let us consider the opposite descending functor $G : (\mathbf{D} \times \mathbf{D})^{\mathbf{1}} \to \mathbf{D}$ with *the dual concept* of the couniversal arrow "*from functor G to c*" (with opposite arrows w. r. t. scheme in (2.13)), represented by the pair (d, g) such that

$$d(a_1) \triangleq (c, c),$$

represented by the scheme (A.14) in the Appendix, with its reelaborated part here and with

$$d'(a_1) \triangleq (a, b),$$



(2.15)

In the case when G is the coproduct functor, so that

$$\begin{aligned} G(d) &= c + c, \\ G(d') &= a + b, \\ G(\underline{f}) &= k_1 + k_2, \end{aligned}$$

so that

$$g = [\mathrm{id}_c, \mathrm{id}_c],$$

and hence

$$f = g \circ G(\underline{f}) = [k_1, k_2],$$

and we obtain an alternative definition of the commutative diagram (2.31), representing the coproducts by the adjunction of the alternative (more general) diagonal functor $\triangle : \mathbf{D} \to \mathbf{D}^{\mathbf{2}_D}$ where $\mathbf{2}_D$ is the discrete index category composed by two index objects a_1 and a_2, and is right adjoint to it modified functor $G : \mathbf{D}^{\mathbf{2}_D} \to \mathbf{D}$ defined by (2.30).

In this case for the coproducts we can use also the standard definition by functor $+ : \mathbf{D} \times \mathbf{D} \to \mathbf{D}$, from the simple product category $\mathbf{D} \times \mathbf{D}$ as presented in diagram above.

With this, we explained how the standard diagonal functor $\triangle : \mathbf{D} \to \mathbf{D} \times \mathbf{D}$ can be replaced in the large family of general diagonal functors $\triangle : \mathbf{D} \to (\mathbf{D} \times \mathbf{D})^{\mathbf{J}}$, as we explained by scheme (2.5) for the representation of the arrow-diagonal functor $\blacktriangle : \mathbf{D} \to (\mathbf{D} \downarrow \mathbf{D})$; This explains the mathematical importance for the introduction of the more powerful general diagonal functors.

The advantage of using the general diagonal functors $\triangle : \mathbf{D} \to \mathbf{C}^{\mathbf{J}}$ is that they are ascending cases in the comma-propagation of functors as well, and hence they are a

consistent part of the comma-propagation theory, important for the extension of the comma-propagation to all (co)limits (provided in next section).

The properties demonstrated previously for the comma-propagation of functors, natural transformations and (co)universal arrows within the infinite hierarchy of n-dimensional levels (arrow categories), are necessary in order to expand the comma-propagation theory to the most useful category structures: adjunctions and (co)limits. Category theory is the theory of the typed composition of arrows and as such is a very "weak" theory. The theory finds its strength in powerful and intricate descriptive mechanisms.

Thus, from the fundamental role of adjunctions and (co)limits, we will show how these important categorial structures, presented in the base categories, propagates by comma-propagation to all of the infinite hierarchy of n-dimensional levels (arrow categories constructed over a given base category).

So, the following generalization of comma-induction is valid for Theorem 2 for the adjunctions.

Proposition 5 (Comma-propagation of adjunctions). *Let $\mathbf{C}^{\mathbf{J}}$ be a category of functors from a small index category $\mathbf{J}$ into a base category $\mathbf{C}$. Then, for every adjunction $(F, G, \varepsilon, \eta)$ where $F : \mathbf{D} \to \mathbf{C}^{\mathbf{J}}$ is left adjoint of a functor G with η and ε the unit and counit of this adjunction, for each n-dimensional level $\mathbf{C}_n$, $n \geq 1$, there exists an adjunction between comma-propagated functors, $(F_n, G_n, \varepsilon_n, \eta_n) : \mathbf{D}_n \to \mathbf{C}_n^{\mathbf{J}}$.*

Proof. Let $(F, G, \varepsilon, \eta)$ be the adjunction in Definition 58 in Section A.2, with the counit $\varepsilon : FG \xrightarrow{\cdot} \mathrm{Id}_{\mathrm{dom}(G)}$ and unit $\eta : \mathrm{Id}_{\mathrm{dom}(F)} \xrightarrow{\cdot} GF$ and with the commutative diagrams in (A.16).

Let us assume, by comma-propagation, that there is the adjunction $(F_n, G_n, \varepsilon_n, \eta_n)$ for $n = k$.

We have to prove that in this adjunction there exists for $n = k + 1$ as well, with comma-induced natural transformation $\eta_{k+1} : \mathrm{Id}_{\mathrm{dom}(F_{k+1})} \xrightarrow{\cdot} G_{k+1}\{F\}_{k+1}$ (and analogously for the counit ε_{k+1}) such that for any object $J(k)$ in $\mathbf{D}_{k+1}$, with an arrow $k : c_1 \to c_2$ in $\mathbf{D}_k$, we obtain the universal arrow $(F_{k+1}(J(k)), \eta_{k+1}(J(k)))$, from $J(k)$ to G_{k+1}. In our case, the functor $F : \mathbf{D} \to \mathbf{C}^{\mathbf{J}}$ is an ascending functor, so that

$$F_{k+1} = L \circ \widehat{F}_k,$$

and $G : \mathbf{C}^{\mathbf{J}} \to \mathbf{D}$ is descending, so that

$$G_{k+1} = \widehat{G}_k \circ K$$

and natural transformation η is a basic case (between two endofunctors of the category $\mathbf{D}$), and hence

$$\eta_{k+1} = \widehat{\eta}_k.$$

From the fact that G is a descending case, we can use the method in the proof of Theorem 3, and consider the universal arrow $(d,(g_1;g_2))$ from $d = F_{k+1}(J(k)) = L^0J(F^1_k(k))$ to the comma-propagated functor $G_{k+1} : \mathbf{C}^{\mathbf{J}}_{k+1} \to \mathbf{D}_{k+1}$.

So, we have from Theorem 3 the corresponding diagram for the arrows in the comma lifted category $\mathbf{D}_{k+1} = (\mathbf{D}_k \downarrow \mathbf{D}_k)$, with

$$d = L^0J(F_k(k)),$$

and $g_i = \eta_k(c_i)$ for $i = 1,2$, i. e.,

$$\begin{aligned} g = (g_1;g_2) &= (\eta_k(c_1);\eta_k(c_2)) \\ &= \widehat{\eta}_k(J(k)) \\ &= \eta_{k+1}(J(k)), \end{aligned}$$

for the comma-propagated universal arrow,

$$(d,g) = (d,(g_1;g_2)) = (L^0J(F_k(k)),(\eta_k(c_1);\eta_k(c_2))), \tag{2.16}$$

we obtain

$$\begin{array}{ccccc} J(k) \xrightarrow{(g_1;g_2)} \widehat{G}_k(J(F_k(k))) & & J(F_k(k)) = K^0(d) & & d = L^0J(F_k(k)) = F_{k+1}(J(k)) \\ \Big\downarrow f = (f_1;f_2) & \Big\downarrow \widehat{G}_k(f_{_1};f_{_2}) & \Big\downarrow & \Big\downarrow (f_{_1};f_{_2}) = K^1(f_{_}) & \Big\downarrow f_{_} \\ G_{k+1}(d') = \widehat{G}_k(J(h)) & & J(h) = K^0(d') & & d' \end{array} \tag{2.17}$$

$$\begin{array}{ccccc} \mathbf{D}_{k+1} & \xleftarrow{\widehat{G}_k} & (\mathbf{C}^{\mathbf{J}}_k \downarrow \mathbf{C}^{\mathbf{J}}_k) & \xleftarrow{K} & \mathbf{C}^{\mathbf{J}}_{k+1} \\ \mathbf{D}_{k+1} & \xrightarrow{\widehat{F}_k} & (\mathbf{C}^{\mathbf{J}}_k \downarrow \mathbf{C}^{\mathbf{J}}_k) & \xrightarrow{L} & \mathbf{C}^{\mathbf{J}}_{k+1} \end{array}$$

so that for each pair (d',f) where the object d' in $(\mathbf{D} \downarrow \mathbf{D})^{\mathbf{J}}$ is a functor $d' = F' : \mathbf{J} \to \mathbf{C}_{k+1}$, and hence, from point 2 of Definition 7, we can define

$$K^0(d') = K^0(F') = J(h),$$

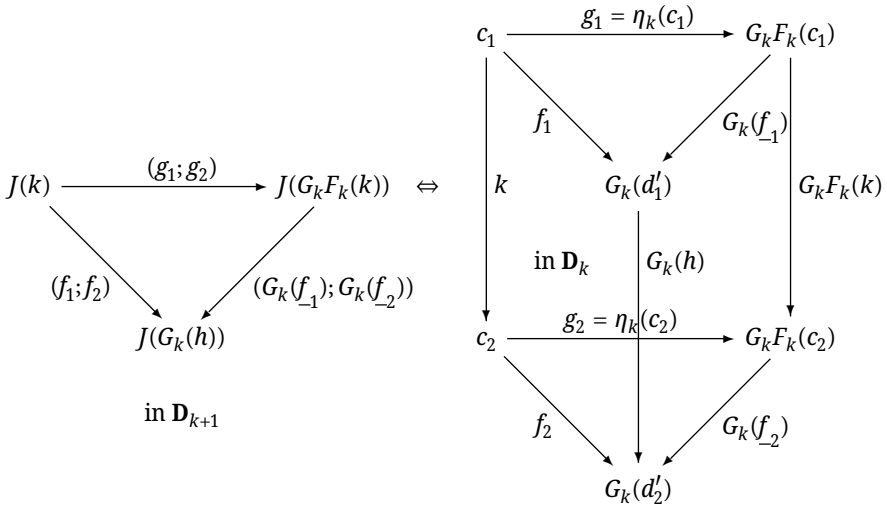
where $h : d'_1 \xrightarrow{\cdot} d'_2$ is a natural transformation between functors $H_1 = d'_1$ and $H_2 = d'_2$ in $\mathbf{C}^{\mathbf{J}}$ that must satisfy (2.10) for each indexed object a_j of small indexed category $\mathbf{J}$, i. e.,

$$h(a_j) = \psi(F'(a_j)) = \psi(d'(a_j)).$$

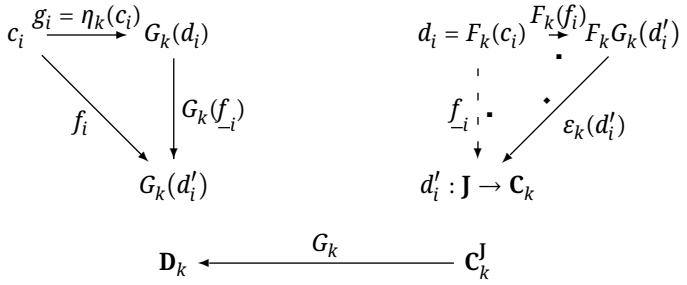
For the left-hand diagram above, we have that

$$\widehat{G}_k(J(F_k(k))) = J(G_kF_k(k)) = \langle G_kF_k(c_1), G_kF_k(c_2), G_kF_k(k)\rangle,$$
$$\widehat{G}_k(J(h)) = J(G_k(h)) = \langle d_1', d_2', h\rangle \quad \text{and}$$
$$\widehat{G}_k(\underline{f}_{-1};\underline{f}_{-2}) = (G_k(\underline{f}_{-1}); G_k(\underline{f}_{-2})),$$

so that the commutative diagram in $\{\mathbf{D}_k\} = \mathbf{D}_{k+1}$ corresponds to the following right-hand side commutative diagram in $\mathbf{D}_k$,



Two commutative triangles in the right-hand side correspond to the following two commutative adjoint diagrams:



(2.18)

i. e., we have that there exist universal arrow, the pair (d_i, g_i) with arrow $g_i = \eta_k(c_i)$, such that for each pair (d_i', f_i), $i = 1, 2$, with objects d_i in $\mathbf{C}_k^{\mathbf{J}}$ (i. e., the functors $d_i' : \mathbf{J} \to \mathbf{C}_k$), there exists a unique arrow $\underline{f}_i$ (a natural transformation between two functors in $\mathbf{C}_k^{\mathbf{J}}$), such that the diagram above commutes. Thus, the arrow $\underline{f} = L(\underline{f}_{-1};\underline{f}_{-2})$ in scheme (2.17) is unique as well.

It is easy to verify (see the diagram (2.17) above) that the comma-propagated adjunction $(F_{k+1}, G_{k+1}, \varepsilon_{k+1}, \eta_{k+1})$ is a composition of the comma-lifting adjunction provided in

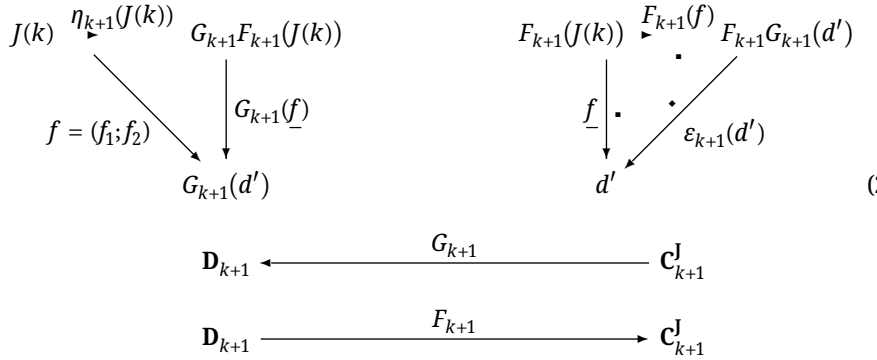
Theorem 1, $(\widehat{F}_k, \widehat{G}_k, \widehat{\varepsilon}_k, \widehat{\eta}_k)$ and the adjunction $(L, K, \mathrm{id}_{LK}, \mathrm{id}_{KL})$ provided by Proposition 3. From the fact that

$$\begin{aligned}\widehat{G}_k(J(F_k(k))) &= \widehat{G}_k(K^0(d)) \\ &= G_{k+1}(d) \\ &= G_{k+1}F_{k+1}(J(k)),\end{aligned}$$

and

$$\begin{aligned}\widehat{G}_k(\underline{f}_1;\underline{f}_2) &= (\widehat{G}_k K)(L(\underline{f}_1;\underline{f}_2)) \\ &= G_{k+1}(\underline{f}),\end{aligned}$$

the diagram (by eliminating the vertical arrow in the middle) in (2.17) is the $(k+1)$-th comma-propagated adjunction,



(2.19)

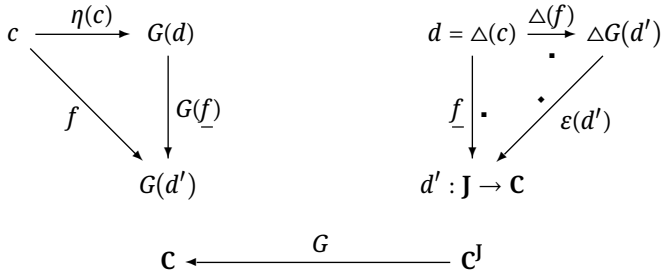
A similar proof can be given for the couniversal arrows corresponding to the counit ε of the adjunction. □

Based on this result of comma-propagation of the adjunctions, and on the result shown in Example 4 that the comma-propagation of the general diagonal functor $\triangle : \mathbf{C} \to \mathbf{C}^{\mathbf{J}}$ preserves its diagonal functor properties, and from the fact that it will be used for consideration of the (co)limits in the n-dimensional hierarchy, we can show this comma-propagation of the (co)limit structures.

Theorem 4. *Let $\mathbf{C}^{\mathbf{J}}$ be a category from a small index category $\mathbf{J}$ into a given base category $\mathbf{C}$, such that for any object (functor) $d' : \mathbf{J} \to \mathbf{C}$ in $\mathbf{C}^{\mathbf{J}}$, there exists a limit $G(d')$ in $\mathbf{C}$ generated by the functor $G : \mathbf{C}^{\mathbf{J}} \to \mathbf{C}$ right adjoint to the general diagonal functor $\triangle : \mathbf{C} \to \mathbf{C}^{\mathbf{J}}$.*

Consequently, for every functor $d' : \mathbf{J} \to \mathbf{C}_n$ which is an object in the category of functors $\mathbf{C}^{\mathbf{J}}_n$, for $n \geq 2$, there exists a limit $G_n(d')$ in $\mathbf{C}_n$, where $G_n : \mathbf{C}^{\mathbf{J}}_n \to \mathbf{C}_n$ is the comma-propagated functor of G and right-adjoint to the comma-propagated diagonal functor $\triangle_n : \mathbf{C}_n \to \mathbf{C}^{\mathbf{J}}_n$.

Proof. By using Propositions 4 and 5 with adjunction $(\triangle, G, \varepsilon, \eta)$ where $\triangle : \mathbf{D} \to \mathbf{C}^{\mathbf{J}}$ is general diagonal functor left-adjoint of a functor G its right-adjoint functor G with η and ε the unit and counit of this adjunction, so that, *for any object* c in $\mathbf{C}$, we have the universal arrow, i. e., pair (d, g) where $d = (d^0, d^1) = \triangle^0(c)$ is the constant functor from a small index category $\mathbf{J} = \mathbf{Sch}(\text{graph})$ (derived as a sketch category from a finite graph, which in this case represents the structure of the diagram of the limit cones) into $\mathbf{C}$ and $g = \eta(c)$, such that for each pair (d', f) there is a unique arrow (natural transformation) $\underline{f} : d \xrightarrow{\cdot} d'$ such that the following diagram commutes:



(2.20)

where the object (functor) $d' : \mathbf{J} \to \mathbf{C}$ in $\mathbf{C}^{\mathbf{J}}$ represents the diagram for which the object $G(d')$ in $\mathbf{C}$ is the limit (i. e., *G creates the limit of the diagram* d').

From the fact that for each index object a_j in $\mathbf{J} = \mathbf{Sch}(\text{graph})$, we have for the constant functor $d = \triangle(c)$, from (2.3), $d^0(a_j) = c$ and for each arrow l_{jm} in $\mathbf{J}$, $d^1(l_{jm}) = \text{id}_c$ from Definition 6 of general diagonal functor, it holds that the diagram represented by this functor $d = \triangle(c)$ is composed by all objects equal to c and all arrows equal to its identity arrow id_c, so that the diagram represented by this functor $d = \triangle(c)$ can be reduced to the single object c, which is a vertex of the cone composed by arrow components $\underline{f}(a_j) : c \to d'(a_j)$ for each index object a_j in $\mathbf{J}$, where $d'(a_j)$ is an object of the diagram of the graph represented by the functor $d' : \mathbf{J} \to \mathbf{C}$.

That is, for each object c in $\mathbf{C}$, the arrow (a natural transformation) $\underline{f} : d \xrightarrow{\cdot} d'$ represents a cone with vertex equal to object c toward the diagram represented by the functor d' (the arrow components $\underline{f}(a_j) : c \to d'(a_j)$ for each index-object a_j in $\mathbf{J}$ generate this cone). The *limit cone* is represented by the natural transformation $\varepsilon(d')$ derived from the counit ε for the given diagram (a functor) d', with the vertex equal to the limit $G(d')$ of this diagram d' (from the fact that the functor $\triangle(G(d'))$ reduces to the single object $G(d')$ in $\mathbf{C}$ as explained previously for the diagonal functor $\triangle : \mathbf{C} \to \mathbf{C}^{\mathbf{J}}$).

From the fact that the comma-propagation preserves the adjunctions, the diagram above holds for each comma-propagated adjunction $(\triangle_n, G_n, \varepsilon_n, \eta_n) : \mathbf{C}_n \to \mathbf{C}_n^{\mathbf{J}}$, for $n \geq 1$, it holds that the limits in $\mathbf{C}$ are comma-propagated to all higher n-dimensional levels $\mathbf{C}_n$.

The proof for the dual case of the *colimits* is analog to this done above, by considering that in this way all arrows in the commutative diagrams in (2.20) *are inverted.* □

Finally, we obtain the following closure property for the n-dimensional levels.

Corollary 4. *Let* $\mathbf{C}$ *be a small-complete category. Then, every n-dimensional level* $\mathbf{C}_n$*, for* $n \geq 2$*, is a small-complete category as well.*

Proof. Directly from Theorem 4, by considering every small index category (derived from a finite diagram) $\mathbf{J}$. □

It is well known that a category $\mathbf{C}$ is a small complete if every its finite diagram has a limit: the enough condition that it is a small complete is that $\mathbf{C}$ has arbitrary finite products and every pair of its arrows has an equalizer.

Thus, these two particular limits, the products and the equalizers have particular importance, and in next section we will discuss in details the comma-propagation of these two limit types in the infinite hierarchy of the arrow categories of a given base small-complete category $\mathbf{C}$. One of the well-known categories with such properties is the **Set** category.

2.4 Example: Comma-propagation and infinite hierarchy of small-complete categories

In this chapter, dedicated to the infinite hierarchy of the arrow categories for a given base category $\mathbf{C}$, we will provide the significant example when $\mathbf{C}$ is a small-complete category, that is having arbitrary finite products and equalizers for each pair of its morphisms.

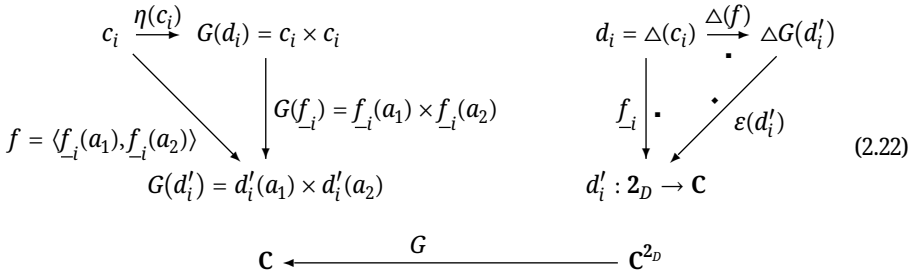
Comma-propagation of the products

Let us now consider an example for the comma-propagation of the limits, partially considered in previous Example 5.

In this case for the limits, we use the adjunction $(\triangle, G, \varepsilon, \eta)$, presented in the proof of Theorem 4, where functor $G = (G^0, G^1) : \mathbf{C}^{\mathbf{J}} \to \mathbf{C}$ is the right adjoint to the general diagonal functor $\triangle : \mathbf{C} \to \mathbf{C}^{\mathbf{J}}$ where, in this case of product limits, the small index category $\mathbf{J} = \mathbf{2}_D$ is the discrete category composed by two indexed objects a_1 and a_2 with their identity arrows, and this functor G is defined by: For each object (a functor) $d_i = (d_i^0, d_i^1) : \mathbf{2}_D \to \mathbf{C}$ and arrow (a natural transformation) $f_{_i} : d_i \xrightarrow{\cdot} d_i'$ in $\mathbf{C}^{\mathbf{2}_D}$,

$$G^0(d_i) \triangleq d_i^0(a_1) \times d_i^0(a_2) \quad \text{and} \quad G^1(f_{_i}) \triangleq f_{_i}(a_1) \times f_{_i}(a_2) \tag{2.21}$$

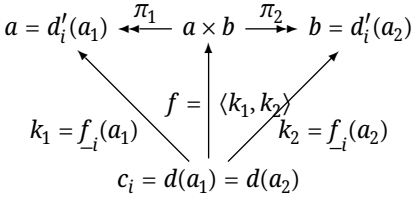
so that for any object $c_i \in \mathrm{Ob}_{\mathbf{C}}$ and the object (functor) d_i' in $\mathbf{C}^{\mathbf{2}_D}$ representing a diagram with $a = d_i'(a_1)$ and $b = d_i'(a_2)$ for which we define the limit cone with its vertex equal to the created limit $G^0(d_i') \triangleq d_i'(a_1) \times d_i'(a_2) = a \times b$ in $\mathbf{C}$, the adjunction used for these limits (the products) in scheme (2.20), becomes equal to the following commutative diagrams:



(2.22)

where $\eta(c_i) = \langle \mathrm{id}_{c_1}, \mathrm{id}_{c_1} \rangle$. Note that from the property of the diagonal functor, we have that $d_i^0(a_j) = \triangle(c_i)(a_j) = c_i$, for $j = 1, 2$, so that $G(d_i) = c_i \times c_i$. So that we have that $k_i \triangleq \underline{f}_i(a_j) : d_i(a_j) \to d'_i(a_j)$ for $j = 1, 2$ are the two arrows $k_1 : c_i \to a$ and $k_2 : c_i \to b$ in $\mathbf{C}$. So that we obtain the standard arrows for products, $f = \langle k_1, k_2 \rangle : c_i \to a \times b$ and $G(\underline{f}_i) = k_1 \times k_2 : c_i \times c_i \to a \times b$ of the left commutative diagram in $\mathbf{C}$ of the scheme above.

Let us show how the natural transformation $\varepsilon(d'_i)$, derived from the counit ε, defines the first and second projections π_1 and π_2, which are the arrows of the limit cone with vertex (limit) $(\triangle G(d'_i))(a_j) = (\triangle(a \times b))(a_j) = a \times b$, for $j = 1, 2$ into the diagram's objects $d'_i(a_1) = a$ and $d'_i(a_2) = b$, so that $\pi_j = \varepsilon(d'_i)(a_j) : (\triangle G(d'_i))(a_j) \to d'_i(a_j)$ for $j = 1, 2$ are well-known projections $\pi_1 : a \times b \to a$ and $\pi_2 : a \times b \to b$, so that the commutative diagrams of vertical composition of natural transformation (right-hand side commutative diagram above) $\underline{f}_i = \varepsilon(d'_i) \bullet \triangle(f)$, with $k_j = \underline{f}_i(a_j)$ cone arrows into diagram represented by functor $d'_i : \mathbf{2}_D \to \mathbf{C}$ and $(\triangle(f))(a_i) = f = \langle k_1, k_2 \rangle$, is represented by the following commutative diagram in $\mathbf{C}$ for the products $a \times b$ (see (A.14) in the Appendix as well), representing the cone of $\underline{f}_i$, (k_1, k_2) with vertex in c_i, and limit cone of $\varepsilon(d'_i)$. (π_1, π_2) with vertex in $a \times b$,



(2.23)

where $\pi_1 : a \times b \twoheadrightarrow a$ and $\pi_2 : a \times b \twoheadrightarrow b$ are the first and second projection (epimorphisms) of the product $a \times b$, and the *unique* arrow $f = (\triangle(f))(a_i)$ from the $\underline{f}_i$ cone into the limit cone.

From the fact that first comma-projection transformations $\triangle_1 = \triangle$ and $G_1 = G$, let us now consider the comma-propagation of this adjunction into the adjunction

$$(\triangle_2, G_2, \varepsilon_2, \eta_2) : (\mathbf{C} \downarrow \mathbf{C}) \to (\mathbf{C} \downarrow \mathbf{C})^{\mathbf{2}_D}$$

where $G_2 = \widehat{G} \circ K$ (descending case), $\triangle_2 = L \circ \widehat{\triangle}$ (ascending case) and basic comma-propagated unit $\eta_2 = \widehat{\eta}$.

As anticipated in the proof of Proposition 5, for an object $J(k)$ in $\mathbf{C} \downarrow \mathbf{C}$ of a given arrow $k : c_1 \to c_2$ in $\mathbf{C}$, the comma-propagated universal arrow in (2.16), when the left adjoint functor $F = \triangle$, is equal to

$$\begin{aligned}(d, g) &= \big(d, (g_1; g_2)\big) \\ &= \big(LJ(\triangle(k)), (\eta(c_1); \eta(c_2))\big)\end{aligned}$$

where

$$\begin{aligned}LJ(\triangle(k)) &= L(J\triangle^1\psi)J(k) \quad \text{from identity } \psi J \\ &= L\widehat{\triangle}J(k) \quad \text{from comma lifting of } \triangle \\ &= \triangle_2(J(k))\end{aligned}$$

so, from the fact that $K \circ L$ is an identity functor, we obtain

$$K(\triangle_2(J(k))) = J(\triangle(k)) \tag{2.24}$$

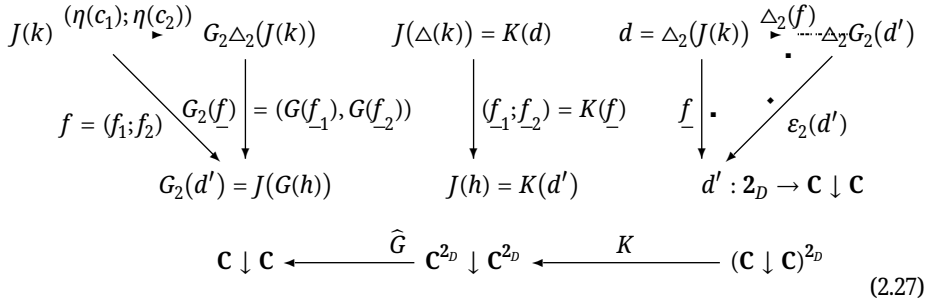
and hence, from the fact that comma-propagated functor $G_2 = \widehat{G}K$, from (2.24) we obtain

$$G_2\triangle_2(J(k)) = \widehat{G}J(\triangle(k)) = J(G(\triangle(k))) = \langle G\triangle(c_1), G\triangle(c_1), G(\triangle(k))\rangle \tag{2.25}$$

and, for an arrow $h \triangleq \psi K^0(d') : d'_1 \to d'_2$ in $\mathbf{C}^{\mathbf{2}_D}$ we obtain

$$G_2(d') = \widehat{G}K(d') = \widehat{G}(J(h)) = J(G(h)) \tag{2.26}$$

So, we have the following comma-propagated adjunction, derived from (2.17) when $F = \triangle$: for each pair (d', f) where $f = (f_1; f_2) : J(k) \to G_2(d')$ in $\mathbf{C} \downarrow \mathbf{C}$, such that there is a unique arrow $\underline{f}$ in $(\mathbf{C} \downarrow \mathbf{C})^{\mathbf{2}_D}$ for which the following diagrams commute:



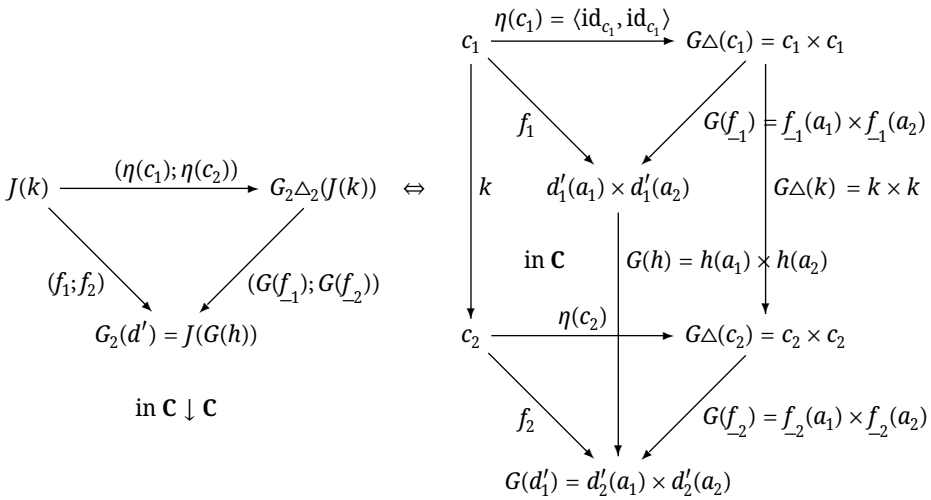
(2.27)

where the vertical arrow in the middle represents the vertical composition of natural transformations (the arrows in $\mathbf{C}^{\mathbf{2}_D}$),

$$\underline{f}_2 \bullet \triangle(k) = h \bullet \underline{f}_1. \tag{2.28}$$

Note that from (2.25) and (2.26) the commutative diagrams above represent the comma-propagated adjunction between comma-propagated functors $\triangle_2$ and G_2. In fact, the commutative diagram in the arrow category $\mathbf{C}_2 = \mathbf{C} \downarrow \mathbf{C}$ and the vertical arrow $(\underline{f}_{-1};\underline{f}_{-2})$ in the arrow category $\mathbf{C}^{2_D} \downarrow \mathbf{C}^{2_D}$ are composed by the two universal arrow diagrams in scheme (2.22), for $i = 1,2$, from the fact that $\widehat{G}(\underline{f}_{-1},\underline{f}_{-2}) = (G(\underline{f}_{-1}); G(\underline{f}_{-2}))$, with the cone $\underline{f}_{-i}$ from the vertex c_i into the diagram (functor) d'_i, and the *limit cone* $\varepsilon(d'_i)$ from product limit (vertex) $G(d'_i)$ into the diagram (functor) d'_i, in the two cases $i = 1,2$.

Thus, the left-hand side commutative diagram in $\{\mathbf{C}\} = \mathbf{C} \downarrow \mathbf{C}$ corresponds to the following right-hand side commutative diagram in $\mathbf{C}$,



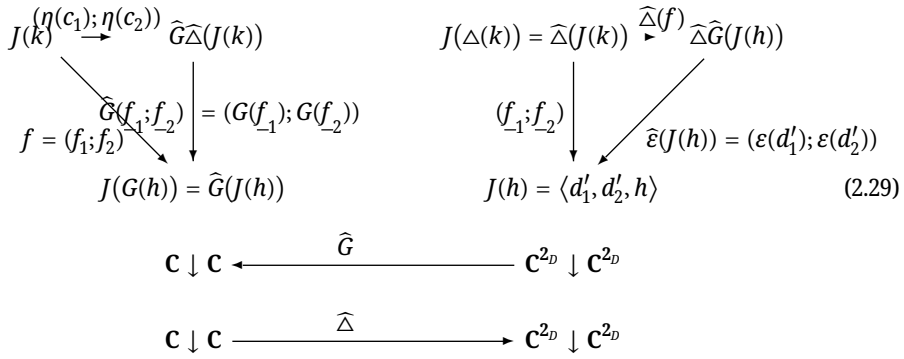
where $f_i = G(\underline{f}_{-i}) \circ \eta(c_i) = \langle \underline{f}_{-i}(a_1), \underline{f}_{-i}(a_2) \rangle$ for $i = 1,2$, and we can see how the comma-propagated limits are composed by the limits of lower level:

$$G_2(d') = J(G(h)) = \langle d'_1(a_1) \times d'_1(a_2), d'_2(a_1) \times d'_2(a_2), h(a_1) \times h(a_2) \rangle.$$

The meaning of the commutative diagram on the right-hand side of scheme (2.27) in the functor category $(\mathbf{C} \downarrow \mathbf{C})^{2_D}$ can be easily understood as follows, by using the comma-lifted adjunction:

$$(\widehat{\triangle}, \widehat{G}, \widehat{\varepsilon}, \widehat{\eta})$$

of the comma-lifted functors $\widehat{\triangle}$ and $\widehat{G}$ whose following diagram of adjunction, by considering that $\widehat{\eta}(J(k)) = (\eta F^0_{\mathrm{st}}; \eta S^0_{\mathrm{nd}})(J(k)) = (\eta F^0_{\mathrm{st}}; \eta S^0_{\mathrm{nd}})(\langle c_1, c_2, k \rangle) = (\eta(c_1); \eta(c_2))$ and analogously that $\widehat{\varepsilon}(J(h)) = \widehat{\varepsilon}(\langle d'_1, d'_2, h \rangle) = (\varepsilon(d'_1); \varepsilon(d'_2))$ and $\widehat{G}(\underline{f}_{-1};\underline{f}_{-2}) = (G^1F^1_{\mathrm{st}}; G^1S^1_{\mathrm{nd}})(\underline{f}_{-1};\underline{f}_{-2}) = (G(\underline{f}_{-1}); G(\underline{f}_{-2}))$, is composed by two adjoint diagrams,



(2.29)

where the left commutative diagram is identical to the left commutative diagram in (2.27) from the fact that

$$\begin{aligned}\widehat{G}\widehat{\triangle}(J(k)) &= \widehat{G}(J\triangle^1\psi)(J(k)) = \widehat{G}(J(\triangle(k))) \quad \text{from identity } \psi J\\ &= J(G(\triangle(k)))\end{aligned}$$

and the right commutative diagram is an extension of the vertical arrow in the middle of scheme (2.27), which satisfies equation (2.28). It is easy now to show that the commutative diagram in the right-hand side of scheme (2.27) is equal to the application of the modulator functor L to the right-hand side commutative diagram in (2.29): in fact, for its objects we have:

1. $d' = LJ(h)$
2. $\triangle_2(J(\mathrm{id}_c)) = L\widehat{\triangle}(J(\mathrm{id}_c))$
3. $\triangle_2\{G\}_2(d') = (L\widehat{\triangle})(\widehat{G}K)(d') = L\widehat{\triangle}\widehat{G}K(L(J(h))) = L\widehat{\triangle}\widehat{G}(J(h))$,

from identity KL, and for its arrows:

1. $\underline{f} = L(\underline{f}_{-1};\underline{f}_{-2})$
2. $\triangle_2(f) = L\widehat{\triangle}(f)$, where $\widehat{\triangle}(f) = \widehat{\triangle}(f_1;f_2) = (\triangle(f_1);\triangle(f_2))$
3. $\varepsilon_2(d') = L\widehat{\varepsilon}K(d')$ because ε is a balanced natural transformation case
   $= L\widehat{\varepsilon}K(LJ(h))$
   $= L\widehat{\varepsilon}(J(h))$, from identity KL.

Note that the comma-propagated adjunction is a composition of the comma-lifted adjunction in scheme (2.29) and the adjunction of L and K (which is right adjoint to L) provided by Proposition 3.

Comma-propagation of the coproducts

Let us now consider an example for the comma-propagation of the colimits.

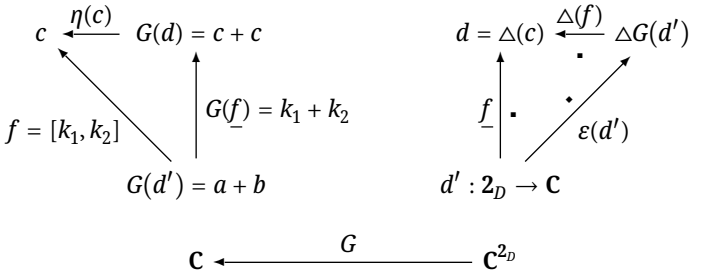
In this case for the colimits, we use the adjunction $(\triangle, G, \varepsilon, \eta)$ with inverted arrows (natural transformations) for the unit and counit, where functor $G = (G^0, G^1) : \mathbf{C}^{\mathbf{J}} \to \mathbf{C}$ is right adjoint to the general diagonal functor $\triangle : \mathbf{C} \to \mathbf{C}^{\mathbf{J}}$ with *inverted*

$$\text{unit } \eta : G\triangle \xrightarrow{\cdot} \text{Id}_{\mathbf{C}} \quad \text{and} \quad \text{countit } \varepsilon : \text{Id}_{\mathbf{C}^{\mathbf{J}}} \xrightarrow{\cdot} \triangle G.$$

In this case of the coproduct, the small index category $\mathbf{J} = \mathbf{2}_D$ is the discrete category composed by two indexed objects a_1 and a_2 with their identity arrows, and this functor G is defined by: For each object (a functor) $d = (d^0, d^1) : \mathbf{2}_D \to \mathbf{C}$ and arrow (a natural transformation) $\underline{f} : d' \xrightarrow{\cdot} d$ in $\mathbf{C}^{\mathbf{2}_D}$,

$$G^0(d) \triangleq d^0(a_1) + d^0(a_2) \quad \text{and} \quad G^1(\underline{f}) \triangleq \underline{f}(a_1) + \underline{f}(a_2) \tag{2.30}$$

so that for any object $c \in \text{Ob}_{\mathbf{C}}$ and the object (functor) d' in $\mathbf{C}^{\mathbf{2}_D}$ representing a diagram with $a = {d'}^0(a_1)$ and $b = {d'}^0(a_2)$ for which we define the colimit $G^0(d') \triangleq {d'}^0(a_1) + {d'}^0(a_2) = a + b$ in $\mathbf{C}$, the adjunction used for these colimits (the coproducts) in scheme (2.20), becomes equal to the following commutative diagrams (with all *arrows inverted*):

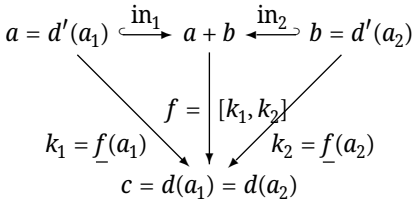


(2.31)

Note that from the property of the diagonal functor, we have that $d^0(a_i) = d^0(a_i) = \triangle(c)(a_i) = c$, for $i = 1, 2$, so that $G(d) = c + c$. So that we have that $k_i = \underline{f}(a_i) : {d'}^0(a_i) \to d^0(a_i)$ for $i = 1, 2$ are the two arrows $k_1 : a \to c$ and $k_2 : b \to c$ in $\mathbf{C}$, so that we obtain the standard arrows for coproducts, $[k_1, k_2] : a + b \to c$ and $k_1 + k_2 : a + b \to c + c$, and $\eta(c) = [\text{id}_c, \text{id}_c] : c + c \to c$ of the left commutative diagram in $\mathbf{C}$ of the scheme above.

Let us show how the natural transformation $\varepsilon(d')$ (an arrow in $\mathbf{C}^{\mathbf{2}_D}$), derived from the counit ε, defines the two monomorphisms in_1 and in_2, which are the arrows of the colimit cocone with vertex (colimit) $(\triangle G(d'))(a_i) = (\triangle(a + b))(a_i) = a + b$, for $i = 1, 2$ from the diagram's objects $d'(a_1) = a$ and $d'(a_2) = b$. Thus, $\text{in}_i = \varepsilon(d')(a_i) : d'(a_i) \to (\triangle G(d))(a_i)$ are well-known injections $\text{in}_1 : a \hookrightarrow a + b$ and $\text{in}_2 : b \hookrightarrow a + b$.

Hence, the commutative diagrams of vertical composition of natural transformations (right-hand side commutative diagram above) $\underline{f} = \triangle(f) \bullet \varepsilon(d')$, with $k_i = \underline{f}(a_i)$ cocone arrows from the diagram are represented by functor $d' : \mathbf{2}_D \to \mathbf{C}$ and $(\triangle(f))(a_i) = f = [k_1, k_2]$, is represented by the following commutative diagram in $\mathbf{C}$ for the coproducts $a + b$ (see (A.15) in the Appendix as well), representing the cocone of $\underline{f}$, (k_1, k_2) with a vertex in c, and a limited cocone of $\varepsilon(d')$. $(\text{in}_1, \text{in}_2)$ with a vertex in $a + b$,

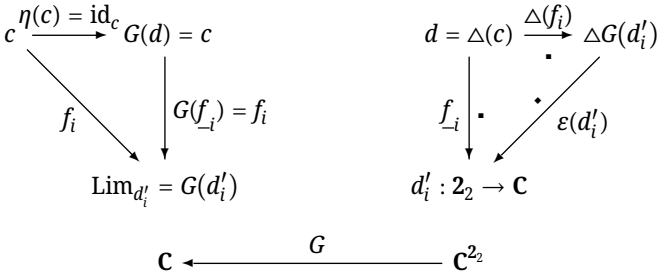


(2.32)

where $\mathrm{in}_1 : a \hookrightarrow a+b$ and $\mathrm{in}_2 : b \hookrightarrow a+b$ are the two injections (monomorphisms) of the coproduct $a+b$, and $f = (\triangle(f))(a_i)$ is the *unique* arrow from the colimit cocone into the $\underline{f}$ cocone.

Comma-propagation of the equalizers

For these kinds of limits, the diagram is a graph composed by two parallel arrows $a_1 \overset{l_{12}}{\underset{l'_{12}}{\rightrightarrows}} a_2$, and hence the sketch (small-index category **J**) obtained from this graph we denote by $\mathbf{2}_2$. So, such diagrams are represented by the category of functors $\mathbf{C}^{\mathbf{2}_2}$, and let $G : \mathbf{C}^{\mathbf{2}_2} \to \mathbf{C}$ be the functor, which creates the limits for such diagrams, and hence right adjoint to the general diagonal functor $\triangle : \mathbf{C} \to \mathbf{C}^{\mathbf{2}_2}$, so that this adjunction is represented by the following commutative diagrams:



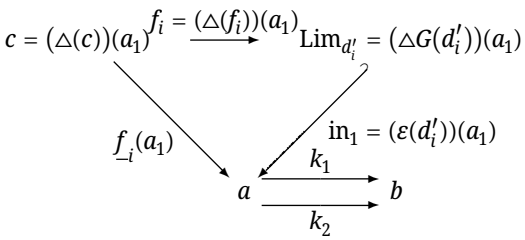
(2.33)

where the functor $d = \triangle(c)$ represents the diagram $c \overset{\mathrm{id}_c}{\underset{\mathrm{id}_c}{\rightrightarrows}} c$, from the fact that $d(a_i) = \triangle(c)(a_i) = c$ and $d(l_{12}) = \triangle(c)(l_{12}) = \mathrm{id}_c$, and hence can be reduced into the object c, i. e., the limit of this diagram is

$$G(d) = c \tag{2.34}$$

so that $\eta(c) = \mathrm{id}_c$, and hence $G(\underline{f}_i) = f_i$ in $\mathbf{C}$, so that the commutative diagram in the left-hand side above reduces to the simple arrow $f_i : c \to \mathrm{Lim}_{d'_i}$. For the commutative diagram in the right-hand side in $\mathbf{C}^{\mathbf{2}_2}$ from the vertex c into diagram $a \overset{k_1}{\underset{k_2}{\rightrightarrows}} b$ (for $a = d'_i(a_1)$, $b = d'_i(a_2)$, $k_1 = d'_i(l_{12})$ and $k_2 = d'_i(l'_{12})$), we have that for a cone $\underline{f}_i$ defined by $\underline{f}_i(a_1)$ into the limit cone $(\triangle G(d'_i))(a_1) = G(d'_i) = \mathrm{Lim}_{d'_i}$ defined by the monomorphism

$\text{in}_1 = (\varepsilon(d_i'))(a_1)$, there is the arrow $f_i = (\triangle(f_i))(a_1)$, which makes this diagram commutative. Hence, from it, we obtain the well-known commutative diagram (see (A.17) in the Appendix) and the monomorphic equalizer $\text{in}_1 = (\varepsilon(d_i'))(a_1) : \text{Lim}_{d_i'} \to a$ in $\mathbf{C}$,



(2.35)

Thus, with the results above, the comma-propagated adjunction for the equalizers is equal to that explained for the products after scheme (2.22), by considering the composition of two commutative diagrams (2.33) for $i = 1, 2$.

2.5 An analogy between physical and abstract categorial Global symmetries: Adjunctions-as-fields

The 1918 article of Emmy Noether [17] contained two theorems, which where *nearly forgotten* within a few years of their publication, for more than 30 years. In the first Noether theorem, she formulated the correspondence between the *symmetries* (nowadays, "the transformation T is a symmetry of the integral I" and "the integral I is invariant under the transformation T" are synonymous) of a variational problem and the *conservation laws* for the associated variational equations. Noether's theorem generated important consequences for quantum mechanics, serving as a guide to the correspondence, which associates conserved quantities to invariances, and it has become the basis for the theory of currents.

We provide, in Section A.4 in the Appendix, a short introduction to the physical symmetries based on Lagrangian densities and the *principle of minimal action* of them in the given time space, both in the Minkowski flat time-space manifold M^4 corresponding to a constant metric tensor field (η_{ij}) given by (A.22), and curved time-space differentiable manifolds $\mathcal{M}$ corresponding to the variable metric tensor field $\mathbf{g}_{jk}$ given by (A.38) used in general relativity in Subsection A.4.1. The transformation of coordinates between two different time-space manifolds is presented in Subsection A.4.2 of the Appendix. In what follows, the "reference frames" correspond to the flat Minkowski time-space coordinate systems.

Analogously, what we demonstrated in last two sections, by examples as well, is the fact that the comma-propagation transformations preserves the (co)limits from base category $\mathbf{C} = \mathbf{C}_1$ to all of its comma-propagated n-dimensional levels $\mathbf{C}_n$, for $n \geq 2$. This invariance of the (co)limit properties in the infinite hierarchy of n-dimensional levels

under the comma-propagation transformation explains why we use the term symmetries in category theory as well.

The Lagrangian formulation, which uses a variational principle to express the equations is very old, with the famous examples of the Maupertious' principle in mechanics and Fermat's principle in optics, and has the following two advantages: as it was emphasized by Dirac and elaborated on by Feynman, it plays a natural role in the path of integral formulation of quantum mechanics. It turns out that the evolution operator for a quantum-mechanical wavefunction can be expressed as a sum over all paths with fixed endpoints in time space weighted by a phase factor that depends only on the Lagrangian integrated along the path. Another advantage is that the Lagrangian, or rather the integral of the Lagrangian density over time and space should be invariant under all symmetries.

Let us try to summarize the results provided in previous sections, in order to analyze better the abstract structure of these symmetries and to compare them in relationship with the well-known symmetry phenomena in physics: we already introduced the interpretation of *adjunctions-as-fields* in Section 2.3. The first analogy to the principle of minimal action of a Lagrangian density in physics, used to derive the *unique* determined trajectory of a material object in the form of its Euler–Lagrange equation of motion, in the category theory of the universal structures as *adjunctions* with the diagonal functors, which define the (co)limits, can be interpreted by the *unique arrow* between any given (co)cone (for a fixed diagram) and limit (co)cone: in this case, the principle of minimal action (in relationship with this (co)limit diagram) determines exactly one arrow for which the adjunction diagram commutes.

In fact, the categorial universal structures, which are invariant under the comma-propagation transformation are the limits and colimits formally specified by general categorial adjunction structures as follows:

– *Limits*: Each limit type in a base category $\mathbf{C}$ (like categorial products, pullbacks, equalizers, etc.) is determined by the kind of limit diagram specified by the small index category (representing a graph of an abstract diagram) $\mathbf{J}$ (see the examples in the previous section). A particular (concrete) limit of such a limit type is specified mathematically by the adjunction structure $(\triangle, G, \varepsilon, \eta)$ between general diagonal functor $\triangle : \mathbf{C} \to \mathbf{C}^{\mathbf{J}}$ and a singular (concrete) functor $G : \mathbf{C}^{\mathbf{J}} \to \mathbf{C}$ with natural transformation (counit) $\varepsilon : \triangle G \stackrel{\cdot}{\longrightarrow} \mathrm{Id}_{\mathbf{C}^{\mathbf{J}}}$ which, for any concrete diagram $d' : \mathbf{J} \to \mathbf{C}$, defines the limit cone $\varepsilon(d') : \triangle G(d') \stackrel{\cdot}{\longrightarrow} d'$ (a natural transformation) from the vertex of this cone, equal to the limit object $G(d')$ in $\mathbf{C}$, toward the objects of this concrete diagram.
  In fact, for each indexed object a_i in $\mathbf{J}$, we obtain an arrow of this limit cone $\varepsilon(d')(a_i) : G(d') \to d'(a_i)$ in $\mathbf{C}$.
– *Colimits*: Each colimit type in a base category $\mathbf{C}$ (like categorial coproducts, pushouts, etc.) is determined by the kind of colimit diagram specified by the small index category (representing a graph of an abstract diagram) $\mathbf{J}$ (see the examples in the

previous section). A particular (concrete) colimit of such a colimit type is specified mathematically by the adjunction structure $(\triangle, G, \varepsilon, \eta)$ between general diagonal functor $\triangle : \mathbf{C} \to \mathbf{C}^{\mathbf{J}}$ and a singular (concrete) functor $G : \mathbf{C}^{\mathbf{J}} \to \mathbf{C}$ with natural transformation (*inverted* counit) $\varepsilon : \mathrm{Id}_{\mathbf{C}^{\mathbf{J}}} \xrightarrow{\cdot} \triangle G$ which, for any concrete diagram $d' : \mathbf{J} \to \mathbf{C}$, defines the limit cocone $\varepsilon(d') : \triangle G(d') \xrightarrow{\cdot} d'$ (a natural transformation) into the vertex of this cone, equal to the colimit object $G(d')$ in $\mathbf{C}$, from the objects of this concrete diagram.
In fact, for each indexed object a_i in $\mathbf{J}$, we obtain an arrow of this colimit cocone $\varepsilon(d')(a_i) : d'(a_i) \to G(d')$ in $\mathbf{C}$.

An analogy between symmetries obtained by transformations in physics (mechanics) and the symmetries, which appear in the comma-propagation transformations in category theory can be given by the following relationship. From the fact that the Lagrangian density $\mathcal{L}$ in physics is defined for a given field $\Psi(p_i)$, where p_i is a time-space point, the first idea is to represent a category $\mathbf{C}$ as discrete "space" where the adjunction (a "field") and an universal arrow of this adjunction for each object of this category is defined. Thus, the objects of the category are analog to the time-space points p_i of this noncontinuous abstract space while the arrows of the category are analog to the paths (so that the composition of two arrows in a category is analogous to the sequential composition of the paths).

1. Let us assume that the categorial analogy to a field Ψ in physics is an adjunction $(\triangle, G, \varepsilon, \eta)$, of a given functor $G : \mathbf{C}^{\mathbf{J}} \to \mathbf{C}$ with the diagonal functor $\triangle : \mathbf{C} \to \mathbf{C}^{\mathbf{J}}$.
2. The analog process in abstract category theory, to derivation of the Euler–Lagrange equation from the Lagrangian density, is the process of formal derivation of the unique arrow from a cone of any given diagram to its limit cone, such that the adjunction's diagram commutes.
3. Symmetries play a fundamental role in physics because they are related to conservation laws. This is stated in Noether's theorem, which says that invariance of the action under a symmetry transformation implies the existence of a conserved quantity (see a *locally conserved* current law (A.31) in Section A.4 in the Appendix). If we consider the *comma-propagation transformation* in this category theory, then the conserved quantities under this transformation can be considered the (co)limit's properties. That is, given a kind of (co)limit in a base category $\mathbf{C} = \mathbf{C}_1$ over a diagram specified by a small indexed category $\mathbf{J}$, and hence defined by the functor $G : \mathbf{C}^{\mathbf{J}} \to \mathbf{C}$, we have shown that by the comma-propagation transformation $\{_\}$ in any n-dimensional level $\mathbf{C}_n$, for $n \geq 2$, we obtain the same type of (co)limits for the fixed diagram in $\mathbf{J}$. In fact, in the previous section, we demonstrated that for a given adjunction in point 1 above, in the next n-dimensional level $\{\mathbf{C}_1\} = \mathbf{C}_2$ we obtain, for a fixed (co)limit diagram by $\mathbf{J}$, *the same* adjunction forms $(\{\triangle\}, \{G\}, \{\varepsilon\}, \{\eta\})$ with only comma-propagated (transformed) functors and natural transformations.

In order to explain these assumptions, we will provide a detailed example, in the first subsection, for derivation of the Lagrangian density of the abstract adjunction field Ψ and how to derive from the principle of minimal action its categorial Euler–Lagrange equation, based on the analogy with the derivations of Einstein's equation from the Einstein–Hilbert action.

Then in the second subsection, we will consider the comma-propagation transformation as an analogy to Galilean boosts global symmetries, and also the fundamental categorial internal symmetry.

We provide the Section A.4 in the Appendix as a short introduction to field theory, the principle of minimal action with derivation of the Euler–Lagrange equation and Noether's conserved current low with two subsections: Subsection A.4.1, as support for the derivation of the Einstein–Hilbert gravitation Lagrange density (the Ricci scalar) in curved time space, and the second Subsection A.4.2 for the coordinate passive transformations.

2.5.1 Analogy between adjunctions and metric tensor field with Einstein–Hilbert action

In GR, the action is usually assumed to be a functional on the metric $\mathbf{g}_{jk}$ given by (A.38) in Section A.4.1 in the Appendix, corresponding to the curved time-space manifold $\mathcal{M}$, and the connection is given by the Levi–Civita connection (as discussed for the non-torsion case after equation (A.39) in Section A.4.1). In 1915, David Hilbert proposed for Lagrangian density $\mathcal{L}_g(\Psi, \partial_l\Psi, p) \triangleq \frac{1}{2\kappa}R$, where $\Psi = \mathbf{g}_{jk}$ is metric tensor field and R is the Ricci scalar given by (A.44) and $\kappa = \frac{8\pi G}{c^4}$ is the Einstein constant, the gravitational part of the action

$$S = \int d\Omega_g \mathcal{L}_g, \tag{2.36}$$

where $d\Omega_g = dq^4\sqrt{-g}$ is an infinitesimal volume in a curved time space. The integral will take over the whole time-space if it converges. If it does not converge, S is no longer well-defined, but a modified definition where one integral is over arbitrarily large, and relatively compact domains, still yields the Einstein equation in GR as the *Euler–Lagrange equation* of the Einstein–Hilbert action.

The derivation of equations from an action allows for easy and direct unification with other *classical* field theories as, e. g., the classical field theory [10] of the particle's complex wave packet.

The Einstein equations in the presence of matter, defined by the Lagrangian density $\mathcal{L}$, (for an elementary particle) are given by adding the matter action to the Einstein–Hilbert action (2.36):

$$S = \int d\Omega_g \mathcal{L}'_g = \int dq^4\sqrt{-g}\left(\frac{1}{2\kappa}R + \mathcal{L}\right) \tag{2.37}$$

where $\mathcal{L}'_g(\mathbf{g}'_{jk}, \partial_i \mathbf{g}'_{jk}, p) = \mathcal{L}_g + \mathcal{L}$ is the resulting Lagrangian of the tensor metric field (gravitational potential) $\mathbf{g}'_{jk}$ obtained as the perturbation of the metric $\mathbf{g}_{jk}$ (and a corresponding new curved time-space manifold $\mathcal{M}'$) because of the presence of matter with Lagrangian density $\mathcal{L}$.

So, the *principle of minimal action* tells us that the variation of this action w. r. t. the inverse metric is zero, i. e.,

$$0 = \delta S = \int dq^4 \left(\frac{1}{2\kappa} \delta(\sqrt{-g}R) + \delta(\sqrt{-g}\mathcal{L}) \right) = \int dq^4 \left(\frac{1}{2\kappa} \frac{\delta(\sqrt{-g}R)}{\delta g^{ij}} + \frac{\delta(\sqrt{-g}\mathcal{L})}{\delta g^{ij}} \right) \delta g^{ij}$$
$$= \int dq^4 \left(\frac{1}{2\kappa} \left(\frac{\delta R}{\delta g^{ij}} + \frac{R}{\sqrt{-g}} \frac{\delta(\sqrt{-g})}{\delta g^{ij}} \right) + \frac{1}{\sqrt{-g}} \frac{\delta(\sqrt{-g}\mathcal{L})}{\delta g^{ij}} \right) \delta g^{ij} \sqrt{-g}.$$

Since this equation holds for any variation δg^{ij}, we obtain the Euler–Lagrange equation of motion,

$$\frac{1}{\kappa} \left(\frac{\delta R}{\delta g^{ij}} + \frac{R}{\sqrt{-g}} \frac{\delta(\sqrt{-g})}{\delta g^{ij}} \right) = -\frac{2}{\sqrt{-g}} \frac{\delta(\sqrt{-g}\mathcal{L})}{\delta g^{ij}} \tag{2.38}$$

where the right-hand side of this equation is (by definition) the stress energy tensor,

$$T_{ij} \equiv -\frac{2}{\sqrt{-g}} \frac{\delta(\sqrt{-g}\mathcal{L})}{\delta g^{ij}} = -2\frac{\delta \mathcal{L}}{\delta g^{ij}} - \mathcal{L} \frac{2}{\sqrt{-g}} \frac{\delta \sqrt{-g}}{\delta g^{ij}},$$

which reduces to[3]

$$T_{ij} = -2\frac{\delta \mathcal{L}}{\delta g^{ij}} + g_{ij}\mathcal{L} \tag{2.40}$$

From the definition of Ricci scalar, by considering that only $\delta g^{ik} \neq 0$,

$$\delta R = \delta \sum_{i,k} g^{ik} R_{ik} = \delta(g^{ik} R_{ik}) = R_{ik} \delta g^{ik} + g^{ik} \delta R_{ik}.$$

The last term $g^{ik}\delta R_{ik}$, multiplied by $\sqrt{-g}$, becomes the total derivative, and thus by the Stokes' theorem only yields a boundary term when integrated. So, when the variation of the metric component δg^{ik} vanishes at infinity, this term does not contribute to the variation of the action. Consequently, we obtain

$$\frac{\delta R}{\delta g^{ij}} = R_{ij}. \tag{2.41}$$

---

3 From the rule of differentiating the inverse of matrix, $\delta g^{ij} = \sum_{k,n} -g^{ik}(\delta g_{kn}) g^{nj}$, we have that $\sum g_{ij} \delta g^{ij} = -\sum g^{ij} \delta g_{ij}$. So, from this result and from the second line in (A.55), by considering that only $\delta g_{ij} \neq 0$, we obtain $\delta\sqrt{-g} = \frac{\sqrt{-g}}{2} g^{ij} \delta g_{ij} = -\frac{\sqrt{-g}}{2} g_{ij} \delta g^{ij}$, i. e.,

$$\frac{1}{\sqrt{-g}} \frac{\delta \sqrt{-g}}{\delta g^{ij}} = -\frac{1}{2} g_{ij}. \tag{2.39}$$

.

Thus, from this result and from (2.39), the Euler–Lagrange equation of motion (2.38) becomes equal to the Einstein equation,

$$\frac{1}{\kappa}\left(R_{ij} - \frac{1}{2}Rg_{ij}\right) = T_{ij}. \tag{2.42}$$

By substitution of T_{ij} from (2.40), we can obtain the direct integration of quantum mechanics and GR by using particle's Lagrangian density given in as explained in [10] (in Section 3.5):

$$R_{ij} - \frac{1}{2}Rg_{ij} = \kappa\left(-2\frac{\delta\mathcal{L}}{\delta g^{ij}} + g_{ij}\mathcal{L}\right), \tag{2.43}$$

which explains how the time-space curvature of the particle's microisland is determined by the particle's field Ψ.

We will use this example from GR as a leading analogy for the categorial symmetry based on the interpretation *adjunctions-as-fields* introduced in Section 2.3 and the principle of minimal action of the categorial scalar field (the adjunctions with diagonal functors), introduced in Section 2.5, corresponding to the existence of the *unique* arrow between any given (co)cone and the (co)limit (co)cone. In what follows, we will consider only the limits. The dual case of the colimits is left as an easy exercise for the riders.

Thus, in category theory we can introduce the following concepts corresponding to the well-known concepts of physics:

1. A scalar field (corresponding to the metric tensor field $\mathbf{g}_{jk}$ of the curved manifold $\mathcal{M}$ in Einstein's GR theory),

$$\Psi(p_i) \triangleq (\triangle, G, \varepsilon, \eta) \tag{2.44}$$

   over the discrete (noncontinuous) space of a category $\mathbf{C}$ composed by the points p_i (the objects of this category) and the oriented paths between them (the arrows of a category). This assumption is explained in the next Section 2.5.2, based on the analysis of the nature of passive comma-propagation transformations and their symmetries. From this analysis, a category $\mathbf{C}$ represents the inertial reference frame (corresponding to Minkowski flat time space in physics) in which we can observe the real curved space of the adjoint category of functors $\mathbf{C}^{\mathbf{J}}$, which defines the particular kind of limits in consideration (in "observation").
2. A Lagrangian density for the field (2.44) at a point $p_i \in \mathrm{Ob}_{\mathbf{C}}$ is the pair

$$\mathcal{L}_g(\Psi, p_i) \triangleq (\triangle^0(p_i), \eta(p_i) : p_i \to G\triangle(p_i)), \tag{2.45}$$

   which is an *universal arrow* of the adjunction at the point p_i. Note that we have no partial derivations $\partial_i\Psi$ as variables of this Lagrangian density, because our space is discrete, and hence we do not have partial derivations.

3. The gravitational part of the action defined by the integral in (2.36) over the space, from the fact that we have a space composed by discrete points $p_i \in \mathrm{Ob}_{\mathbf{C}}$ and directed paths between them, is reduced to path integral given by the set of arrows,

$$S = \int d\Omega_g(\triangle^0(p_i), \eta(p_i)) \triangleq \bigcup_{p_i \in \mathrm{Ob}_{\mathbf{C}}} \{\eta(p_i) : p_i \to G\triangle(p_i)\} \subseteq \mathrm{Mor}_{\mathbf{C}}. \tag{2.46}$$

So, the action integral is reduced to the subset of paths (the arrows of **C**). For a given point p_i, the action in this point $\delta S_{p_i} = \eta(p_i)$ is the displacement between the point p_i in the flat space **C** from the projection $G\triangle^0(p_i)$ (of this point in real curved space $\mathbf{C}^{\mathbf{J}}$ into this flat reference frame **C**).

That is, p_i in the flat space **C** is the position of an observed point if the real space in $\mathbf{C}^{\mathbf{J}}$ is not curved, while from the fact that it is curved, the projected point $G\triangle^0(p_i)$ is the position of this observed point in **C** obtained by the coordinate transformation from the curved manifold of the space $\mathbf{C}^{\mathbf{J}}$ into the flat observer's space **C** (see Section A.4.2 in the Appendix for such transformations of coordinates). Thus, obtained displacement (the path, i. e., arrow $\eta(p_i)$) in **C** is a measure of the observed space curvature at the point p_i.

4. In the presence of matter, which in this categorial framework is represented by the *limit cone* $\varepsilon(d')$ of a given diagram $d' : \mathbf{J} \to \mathbf{C}$. So, its Lagrangian density in the curved space $\mathbf{C}^{\mathbf{J}}$ is, analogously to point 2 above, the (dual) universal arrow $\mathcal{L}'(\Psi', d') = (G(d'), \varepsilon(d'))$ at point d' of $\mathbf{C}^{\mathbf{J}}$, i. e., the arrow (counit) $\varepsilon(d') : \triangle^0(G(d')) \dashrightarrow d'$. However, from the fact that we compute the (total) action in the observer's flat reference frame **C**, this matter, in the *observer's reference frame* **C**, is represented by the Lagrangian density $\mathcal{L}(\Psi, p_i)$, such that *at each* point p_i,

$$\mathcal{L}(\Psi, p_i) = \left(d', G(\underline{f}_i) : G\triangle(p_i) \to G(d')\right) \tag{2.47}$$

for the cone $\underline{f}_i$ from p_i into the diagram d'. Consequently, the resulting Lagrangian $\mathcal{L}'_g = \mathcal{L}_g + \mathcal{L}$ of gravitational field *with matter* in any point p_i in the reference frame **C**, in the categorial framework is represented by the $\mathcal{L}_g(\Psi, p_i)$ equal to the universal arrow (2.45) plus the matter's Lagrangian density in (2.47),

$$\begin{array}{ccccc}
p_i & \xrightarrow{\eta(p_i)} & G\triangle(p_i) & \qquad & \triangle(p_i) \quad \triangle G(d'_i) \\
 & & \Big\downarrow{\scriptstyle G(\underline{f}_i)} & & \Big\downarrow{\scriptstyle \underline{f}_i} \quad \searrow^{\varepsilon(d'_i)} \\
 & & \mathrm{Lim}_{d'_i} = G(d'_i) & & d'_i : \mathbf{J} \to \mathbf{C} \\
 & & \mathbf{C} & \xleftarrow{\qquad G \qquad} & \mathbf{C}^{\mathbf{J}}
\end{array} \tag{2.48}$$

where $\underline{f}_i$ is the cone from point p_i into the diagram d', and $G(\underline{f}_i)$ is the added displacement of the point p_i generated by the changing of the curved space generated by the presence of this matter. Thus, the complete displacement of this point p_i is the composition of the displacement $\eta(p_i)$, generated by the curved space without the presence of the matter, with this added displacement caused by the presence of matter. That is, the action (2.37) in this case is given by

$$S = \int d\Omega_g \mathcal{L}'_g \triangleq \bigcup_{p_i \in \mathrm{Ob}_{\mathbf{C}}} \{G(\underline{f}_i) \circ \eta(p_i) : p_i \to G(d')\} \subseteq \mathrm{Mor}_{\mathbf{C}}. \tag{2.49}$$

Let us consider now the action $S' = \int d\Omega_g \mathcal{L}'_g = S + \delta S$ under perturbation for which, from the principle of minimal action, the variation δS must be zero, and it has to be valid for each point p_i in the space $\mathbf{C}$, i. e.,

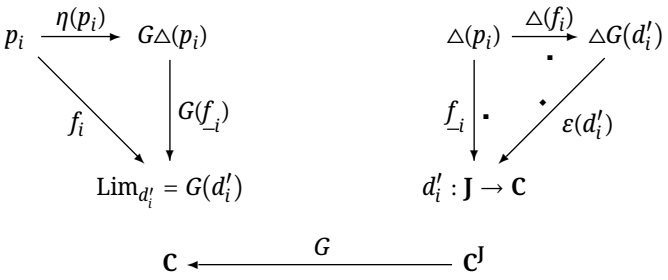
$$S'_{p_i} = \int_{p_i} d\Omega_g \mathcal{L}'_g = S_{p_i} + \delta S_{p_i}, \tag{2.50}$$

where all components of actions in single points in this equation are just the arrows; we denote by the arrow f_i with domain p_i, the arrow in $\mathbf{C}$ corresponding to action S'_{p_i}.

So, from (2.49), $S_{p_i} = G(\underline{f}_i) \circ \eta(p_i)$. Zero actions correspond to the identity arrows (zero displacements) in $\mathbf{C}$, and hence to the action component $\delta S_{p_i} = 0$ corresponds to the identity arrow $\mathrm{id}_{G(d')}$. Thus, their composition (in the right-hand side of equation (2.50)) is just the composition of two arrows $\mathrm{id}_{G(d')} \circ (G(\underline{f}_i) \circ \eta(p_i))$, and hence the equation (2.50) corresponds to the following categorial "Euler–Lagrange equation,"

$$f_i = \mathrm{id}_{G(d')} \circ \big(G(\underline{f}_i) \circ \eta(p_i)\big) = G(\underline{f}_i) \circ \eta(p_i) \tag{2.51}$$

corresponding to the commutative completion of diagram (2.48), i. e.,



(2.52)

representing the commutative diagram of the adjunction. So, the obtained Euler–Lagrange equation corresponds to the commutative diagram of the adjunction in the space $\mathbf{C}$, while the principle of minimal action corresponds to the fact that the arrow f_i is a unique arrow for which the commutative diagram of this adjunction commutes, and this is just the *unique arrow* from the cone $\underline{f}_i$ into the *limit* cone $\varepsilon(d'_i)$.

Consequently, this approach to the analogy between physics and category theories can be summarized as follows (see Table 2.1):

Table 2.1: Adjunction-as-field.

Physics Theory	**Category Theory**
Continuous time space with points p_i	Category **C** as a flat space with objects p_i as points and arrows as directed paths between them
Scalar field Ψ of a given type **J**	Adjunction $(\triangle, G, \varepsilon, \eta)$ with a left-adjoint diagonal functor $\triangle : \mathbf{C} \to \mathbf{C}^{\mathbf{J}}$, of type defined by category **J**
Lagrangian density $\mathcal{L}(\Psi, p_i)$	Universal arrow $(\triangle(p_i), \eta(p_i))$ at a point p_i
Action $S = \int \Omega_g \mathcal{L}(\Psi, p_i)$	Set of displacements of each point p_i (path integrals) represented by arrows with domain objects at p_i
Principle of *minimal* action	Property that for each cone of a diagram d' in $\mathbf{C}^{\mathbf{J}}$ there is a *unique* arrow to the limit cone
Euler–Lagrange equation with position variable p_i of a material object	Commutative diagram in **C** of the adjunction for a fixed diagram d' ("material object") in $\mathbf{C}^{\mathbf{J}}$ and for each variable position p_i

2.5.2 Comma-propagation transformation symmetries

We recall that in physics, a symmetry of a physical system is a physical or mathematical feature of the system (observed or intrinsic) that is preserved or remains unchanged under some transformation. Thus, it represents a kind of some invariance. An important example of such symmetry is the invariance of the form of physical laws under arbitrary differentiable coordinate transformations. Recall that a symmetry is defined as a transformation that keeps the action invariant.

In physics, the Galilean invariance of Newtonian mechanics and the Lorentz invariance of the special theory of relativity are paradigmatic examples of *global* time-space *symmetries*.

As is usually accepted, the Galilean group is the symmetry group of continuous time-space transformations of classical and nonrelativistic quantum mechanics. In the language of the *passive interpretation*, the invariance of the dynamical laws amounts to the equivalence among inertial reference frames (time-translated, space-translated, space-rotated or uniformly moving with respect to each other).

In other words, Galilean transformations do not introduce a modification in the physical situation, but only express a change in the perspective from which the system is described (i. e., from the observer's point of view from his reference frame coordinate system). This merely perspective meaning of the Galilean symmetries depends, in turn,

on the properties of space and time. Invariance under time-translations express the homogeneity of time; invariance under space-translations and space-rotations expresses the homogeneity and the isotropy of space, respectively.

While in Galilean physics we have the flat Minkowski time space with four open infinite dimensions (time and three orthogonal space dimensions) in which there exist and move the material objects, in categorial mathematical theory each n-dimensional level $\mathbf{C}_n$, for $n \geq 1$ with $\mathbf{C}_1 = \mathbf{C}$ equal to the base category with the adjunction-as-field interpretation provided in Table 2.1 in the previous subsection, can be considered as a flat reference frame in which discrete space is composed by the points (corresponding to the objects of the category) and the paths between them (corresponding to the arrows between the objects).

Thus, we will not consider here the active transformations (as, e. g., the Lorentz boosts). In the place of them, we will consider another transformation from one to another reference frame with the constant velocity $\overrightarrow{\mathbf{v}}$ between them ("Galilean boosts"). Though the transformations are named for Galileo, it is absolute time and space as conceived by Isaac Newton that provides their domain of definition.

In essence, the Galilean transformations embody the intuitive notion of addition and subtraction of velocities as vectors. The Galilean boost along the spatial axes with orthonormal basis vectors $\mathbf{e}_j$, $j = 1, 2, 3$, is a passive coordinate transformation of the 4-D time-space points p of a given Minkowski reference frame into another Minkowski reference frame, which moves with constant velocity with spatial components v_j, $j = 1, 2, 3$, w. r. t. the first (observers) frame,

$$p \quad \mapsto \quad p + v_j t \mathbf{e}_j. \tag{2.53}$$

Consequently, in our abstract categorial framework, composed by the base category $\mathbf{C} = \mathbf{C}_1$ and all its comma-propagated n-dimensional levels $\mathbf{C}_n$, for $n \geq 2$, we have such a passive interpretation of Galilean boost transformations from a reference frame $\mathbf{C}_n$ into the reference frame $\mathbf{C}_{n+1}$ by the comma-propagation $\{_\}$ transformation (passive transformation caused by this "constant velocity between them").

Consequently, we consider the symmetries of Minkowski time space (each n-dimensional level $\mathbf{C}_n$ represents a flat Minkowski time space) caused by transformations between uniformly moving frames of reference. So, the physical Galilean boost passive transformation of the coordinates between two inertial frames (2.53) from $\mathbf{C}_n$ into $\mathbf{C}_{n+m}$, in our categorial framework corresponds to the passive transformation operation $\{_\}^m$ equal to m-times application of the one step comma-propagation $\{_\}$ for $m \geq 1$.

Let us consider the one-step comma-propagation transformation $\{_\}^1 = \{_\}$ of a point $p = c$ (an object c of observer's reference frame $\mathbf{C}_n$ having adjunction-as-field $\Psi = (\triangle_n, G_n, \varepsilon_n, \eta_n)$) and an arrow f_i, into the comma-propagation transformed point $\{c\} = J(\mathrm{id}_c)$ in another reference frame $\mathbf{C}_{n+1}$, i. e., to the coordinate and paths transformations defined by Definition 5 in Section 2.1,

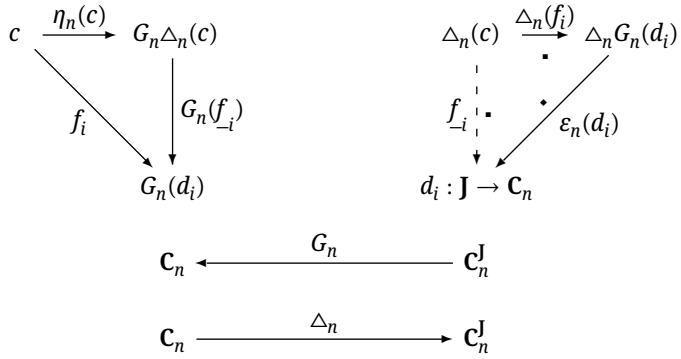
$$c \quad \mapsto \quad \{c\} \triangleq \blacktriangle^0(c) = J(\mathrm{id}_c), \quad \text{and} \quad f_i \quad \mapsto \quad \{f_i\} \triangleq \blacktriangle^1(f_i) = (f_i; f_i) \tag{2.54}$$

from the reference frame $\mathbf{C}_n$ into the reference frame $\mathbf{C}_{n+1}$.

From the Proposition 5 of comma-propagation of the adjunction, in the case when $\mathbf{D} = \mathbf{C}$ and the left-adjoint functor F_n is diagonal functor $\triangle_n$, we have that the comma-propagation transformation $\{_\}$ of adjunction-as-field $\Psi = (\triangle_n, G_n, \varepsilon_n, \eta_n)$ is given by

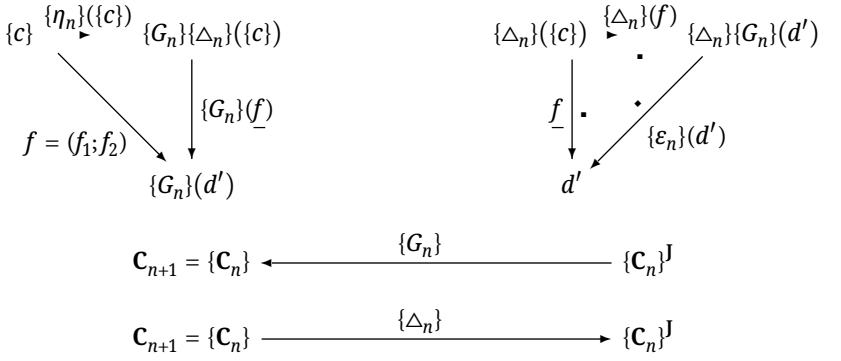
$$\Psi = (\triangle_n, G_n, \varepsilon_n, \eta_n) \quad \mapsto \quad \Psi' = \big(\{\triangle_n\}, \{G_n\}, \{\varepsilon_n\}, \{\eta_n\}\big). \tag{2.55}$$

Thus, the comma-propagated adjunction diagram (2.18) in the case when $k = n$, and $d = \triangle_n(c)$, is reduced to the following commutative diagram for $i = 1, 2$:



(2.56)

so that the comma-propagated adjunction diagram (2.19) is reduced to the following commutative diagram in the particular case when $J(k) = \{c\} = J(\mathrm{id}_c)$, i. e., for the arrow $k = \mathrm{id}_c : c \to c$,



(2.57)

where $\{\eta_n\}(\{c\}) = \widehat{\eta}_n(J(\mathrm{id}_c)) = (\eta_n F^0_{\mathrm{st}}; \eta_n S^0_{\mathrm{nd}})(J(\mathrm{id}_c))$, i. e.,

$$\{\eta_n\}(\{c\}) = \big(\eta_n(c); \eta_n(c)\big) \tag{2.58}$$

and

$$\begin{aligned}\{G_n\}\{\triangle_n\}(\{c\}) &= (\widehat{G}_n K)(L\widehat{\triangle}_n)(J(\mathrm{id}_c))\\ &= \widehat{G}_n\widehat{\triangle}_n(J(\mathrm{id}_c))\\ &= (JG_n^1\psi)(J\triangle_n^1\psi)(J(\mathrm{id}_c))\end{aligned}$$

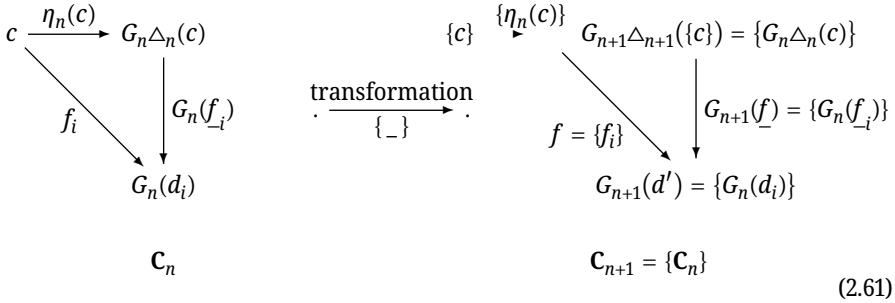
i. e., we obtain the identity arrow in $\mathbf{C}_{n+1}$,

$$\{G_n\}\{\triangle_n\}(\{c\}) = J(G_n^1\triangle_n^1(\mathrm{id}_c)) = \langle G_n\triangle_n(c), G_n\triangle_n(c), G_n^1\triangle_n^1(\mathrm{id}_c)\rangle. \tag{2.59}$$

Thus, the universal arrow $(\triangle_n(c), \eta_n(c))$ in (2.56) is comma-propagated into the universal arrow $(\{\triangle_n\}(\{c\}), \{\eta_n\}(\{c\}))$ in (2.57), and from the fact that they represent the Lagrangian densities, we have the following comma-propagation transformation of the Lagrangian density:

$$\mathcal{L}(\Psi, c) = (\triangle_n(c), \eta_n(c)) \quad \mapsto \quad L'(\Psi', \{c\}) = (\{\triangle_n\}(\{c\}), \{\eta_n\}(\{c\})). \tag{2.60}$$

Remark. In the special case when $d' = L^0(J(\mathrm{id}_{d_i}))$, so that $K(\underline{f}) = (\underline{f}_i;\underline{f}_i)$ with $G_{n+1}(\underline{f}) = \widehat{G}_n K(\underline{f}) = (G_n(\underline{f}_i); G_n(\underline{f}_i)) = \{G_n(\underline{f}_i)\}$, by the comma-propagation transformation $\{_\}$, we obtain the following transformation of the commutative diagram in $\mathbf{C}_n$, presented in the left-hand side in (2.56), into the transformed commutative diagram in the comma-propagated category $\mathbf{C}_{n+1} = \{\mathbf{C}_n\}$, and presented in the left-hand side in (2.57),



(2.61)

i. e., we obtained a reduction of comma-propagation to the "functorial" transformation given by (2.54), from the fact that

$$\begin{aligned}\eta_{n+1}(\{c\}) &= \widehat{\eta}_n(J(\mathrm{id}_c))\\ &= (\eta_n(c); \eta_n(c))\\ &= \{\eta_n(c)\},\end{aligned}$$

and

$$\begin{aligned}G_{n+1}(d') &= \widehat{G}_n K(L(J(\mathrm{id}_{d_i})))\\ &= \widehat{G}_n(J(\mathrm{id}_{d_i}))\\ &= (JG_n^1\psi)(J(\mathrm{id}_{d_i}))\end{aligned}$$

$$= J(G_n^1(\mathrm{id}_{d_i}))$$
$$= J(\mathrm{id}_{G_n(d_i)}) = \{G_n(d_i)\}$$

so that the vertical arrow in the right-hand side $G_{n+1}(\underline{f})$ is a commutative diagram in $\mathbf{C}_n$,

$$\begin{array}{ccc} G_n\triangle_n(c) & \xrightarrow{\mathrm{id}_{G_n\triangle_n(c)}} & G_n\triangle_n(c) \\ \Big\downarrow{\scriptstyle G_n(\underline{f}_i)} & & \Big\downarrow{\scriptstyle G_n(\underline{f}_i)} \\ G_n(d_i) & \xrightarrow{G_n(\mathrm{id}_{d_i})} & G_n(di) \end{array}$$

with $G_n(\mathrm{id}_{d_i}) = \mathrm{id}_{G_n(d_i)}$. With this, we have shown that comma-propagation $\{_\}$ is more general than a functor, because it is not only a functorial transformation of objects and arrows of a given category, equal to arrow-diagonal functor $\blacktriangle$ from (2.54), but a transformation of the functors and natural transformations as well. □

Thus, by one-step comma-propagation transformation from a n-dimensional level to the next $(n+1)$-dimensional level both fields Ψ in (2.55) and its Lagrange-a-field in (2.60) are preserved (remain invariant categorial structures): We only see differently the diagrams in these two different reference frames $\mathbf{C}_n$ and $\mathbf{C}_{n+1}$.

So, it is natural to consider each n-dimensional level $\mathbf{C}_n$, for $n \geq 1$, as a particular reference frame in which categorial structures of the base category $\mathbf{C} = \mathbf{C}_1$ are only differently seen by the observer. For example, let us consider a base category $\mathbf{C} = \mathbf{C}_1$ having the categorial products as universal structures. We have shown that in this case in all n-dimensional levels we will have the categorial products as well because the comma-propagation transformations preserves all (co)limits of the base category. Let an observer be in the n-dimensional level $\mathbf{C}_n$, for $n \geq 2$. If, from this reference system he observes the categorial products in another k-dimensional reference system level for $k \geq 1$ with $k \neq n$, depending on the relationships between these two reference systems (the constant "speed" between them $m = n - k \neq 0$), we obtain that the analogy with Galilei transformations corresponds to the case of *Galilean boost* between uniformly moving frames of reference. That is, the case when this observer sees an inertial propagation (with constant speed) of a material object moving with different speed values in these two different frames of reference.

Thus, this observer in reference frame $\mathbf{C}_n$ can observe the product cones in the base level and do observe that each product cone in his reference frame corresponds to an n-dimensional commutative cube in the base reference frame $\mathbf{C} = \mathbf{C}_1$. That is, each arrow of the cone (first and second projections) of the product in his reference frame is composed by 2^{n-1} arrows in the base reference frame and the objects in his reference frame are composed by the commutative $(n-1)$-dimensional cubes (or a single arrow if $n = 2$) of the base reference frame. This observer for $k > n$ from his reference frame $\mathbf{C}_n$ will observe the product cones in the higher reference frame $\mathbf{C}_k$, but will verify that each

arrow of the product cone in $\mathbf{C}_k$ is composed by $k - n + 1$ arrows of his reference frame and that the objects in other reference frame $\mathbf{C}_k$ correspond to the $(k - n)$-dimensional commutative cubes (or a single arrow if $k - n = 1$) of his reference frame.

Thus, the comma-propagation is a passive transformation, which can be seen as an abstract Galilean boost with reference frames corresponding to the n-dimensional levels $\mathbf{C}_n$ for observation of the same "physical system" corresponding to the base category $\mathbf{C}$. It is a *global symmetry* in the sense that it is not local to a given n-dimensional level but holds in all n-dimensional levels.

3 Arrows-to-objects conceptual transformation: Internal categorial symmetry

3.1 Introduction to categorial internal symmetry of primitive categorial concepts

The main idea of this chapter is based on the consideration of the relationships between the principal *concepts* of the categories, the objects and the arrows (morphisms). It is well known that, in a given category $\mathbf{C}$, its arrows in $\mathrm{Mor}_{\mathbf{C}}$ can be composed by the *partial* operation $\circ : \mathrm{Mor}_{\mathbf{C}} \times \mathrm{Mor}_{\mathbf{C}} \to \mathrm{Mor}_{\mathbf{C}}$ to obtain more complex arrows, so that in the partial algebra $\mathcal{A}_M = (\mathrm{Mor}_{\mathbf{C}}, \circ)$, for each object $a \in \mathrm{Ob}_{\mathbf{C}}$, the identity arrow $\mathrm{id}_a \in \mathrm{Mor}_{\mathbf{C}}$ is the "representation" of this object. From this point of view, the properties of the category can be represented by using *only* its arrows. Such compositional property for the objects and the possibility to represent the category *only* by using the objects of a category generally does not hold. That is, generally we have no this conceptual symmetry (to represent all categorial properties) between the arrows and the objects of a given category.

Our aim is to investigate the class of categories, which are *conceptually symmetric*, i. e., where each arrow can be equivalently represented by some object of this category, and in which we can define the partial operation $* : \mathrm{Ob}_{\mathbf{C}} \times \mathrm{Ob}_{\mathbf{C}} \to \mathrm{Ob}_{\mathbf{C}}$, able to represent the "composition of the objects" of the category. This symmetry appears as an *arrows-to-objects transformation* and is a kind of an *internal symmetry*, called "conceptual symmetry" as well.

Example 6 (Internal symmetries). In physics, we have also a kind of symmetry, different from the translations/rotations in time space as, e. g., in quantum physics for a particle represented mathematically by a complex scalar de Broglie field $\Psi = \Phi(t, \overrightarrow{\mathbf{r}})\mathrm{e}^{-i\varphi_T}$ representing the wave-packet of a free spin-zero neutral particle, where $\Phi(t, \overrightarrow{\mathbf{r}})$ is a real module of this field and φ_T is the De Broglie particle's phase.

For the Lie group G of continuous transformations of the field Ψ, we use the most general one: circle group (unitary group) of transformations, for a given real values α, $U(1) = \{\mathrm{e}^{-i\alpha} \mid \alpha \in \mathbb{R}\}$, with unit 1 (for $\alpha = 0$), multiplication as the group operation, and for each element $\mathrm{e}^{-i\alpha} \in U(1)$ its inverse element $\overline{\mathrm{e}^{-i\alpha}} = \mathrm{e}^{i\alpha}$. From standard quantum mechanics, two kets $|\Psi\rangle$ and $|\mathrm{e}^{-i\alpha}\Psi\rangle$ represent the same particle (it is changed only by De Broglie particle's phase φ_T by the constant value α), so that the particle remains invariant by this general group $U(1)$ of transformations.

This is an *internal* transformation because it transforms the field Ψ into each other in a way without making reference on time and space. In effect, in [10], where $\Phi(t, \overrightarrow{\mathbf{r}})$ is considered as the square root of the energy-density of a given massive particle (fermion) localized for each time-instance t in a small 3D volume V, for a given Lagrangian density $\mathcal{L}_{\mathrm{free}}$ of a free particle propagating by constant speed $\overrightarrow{\mathbf{v}}$ in the vacuum, that the Noether charge density J_0 is equal to the energy-density Φ^2, which remains invariant during such

https://doi.org/10.1515/9783111081670-003

a constant-speed propagation in time space while the Noether current $\overrightarrow{J}$ is just equal to the constant energy-density flux $\Phi^2\overrightarrow{\mathbf{v}}$.

This kind of *internal* categorial symmetry is denominated as a conceptual categorial symmetry of transformation of the arrows into the objects of the category preserving the properties of commutative diagrams (composition of the arrows). A category for which such internal transformation is possible is called a *perfectly symmetric* category, for which there exists the duality operator $B_{\top}$ which, applied to an arrow f of such a category, returns with its "conceptualized object" $\widetilde{f} = B_{\top}(f)$.

It is well known that, given a base category $\mathbf{C}$, we can represent its morphisms as objects by using a derived *arrow* category $\mathbf{C} \downarrow \mathbf{C}$ (a special case of the comma category), such that for a given arrow $f : a \to b$ between two objects $a, b \in \mathrm{Ob}_{\mathbf{C}}$, we have the object denoted by $\langle a, b, f\rangle \in \mathrm{Ob}_{\mathbf{C}\downarrow\mathbf{C}}$ in this arrow category. Thus, in particular cases (for *conceptually closed* categories), we can use a standard categorial language for how to eventually obtain the object in $\mathbf{C}$ representing its arrow f in this way: by *using a functor,* which maps the object $\langle a, b, f\rangle$ in the arrow category $\mathbf{C} \downarrow \mathbf{C}$ into the object in $\mathbf{C}$, denoted by $\widetilde{f}$, able to represent this arrow f. Obviously, this functor, denoted by $T_e : \mathbf{C} \downarrow \mathbf{C} \to \mathbf{C}$, does not exist generally for each category, and our work is to define the subclass of perfectly symmetric categories $\mathbf{C}$ where such a construction is possible. Clearly, the categories that support such functorial "reflection" of arrows into its objects are more expressive and powerful.

This construction is intimately in a relationship with the *commutative diagrams* in the base category $\mathbf{C}$, thus with the *equational theory*, because each arrow in the higher level category $\mathbf{C} \downarrow \mathbf{C}$, corresponds to a commutative diagram in the base category $\mathbf{C}$, with equation $h_2 \circ f = g \circ h_1$, as follows from diagram (3.1) in the case when $n = 1$. Consequently, based on this hierarchy of the n-dimensional levels, we can represent the functorial categorial symmetry in a given n-dimensional level, for $n \geq 0$, by the following diagram:

$$\begin{array}{ccc} a & \xrightarrow{\ f\ } & b \\ {\scriptstyle h_1}\downarrow & & \downarrow{\scriptstyle h_2} \\ c & \xrightarrow{\ g\ } & d \end{array} \quad \Leftrightarrow \quad \begin{array}{c} J(f) = \langle a, b, f\rangle \\ \downarrow{\scriptstyle (h_1, h_2)} \\ J(g) = \langle c, d, g\rangle \end{array} \quad T_e \mapsto \quad \begin{array}{c} \widetilde{f} \\ \downarrow{\scriptstyle T_e(h_1; h_2)} \\ \widetilde{g} \end{array} \tag{3.1}$$

$$\text{in } \mathbf{C}_n \qquad \text{in } \mathbf{C}_{n+1} = \mathbf{C}_n \downarrow \mathbf{C}_n \qquad \text{in } \mathbf{C}_n$$

As we can see, in order to satisfy such a categorial symmetry property, not only the "conceptualized" objects $\widetilde{f}$ and $\widetilde{g}$ (obtained from the respective arrows $f : a \to b$ and $g : c \to d$) must exist in $\mathbf{C}_n$, but they also must coherently represent the rule of these

arrows,[1] and the arrow $T_e(h_1; h_2)$ in $\mathbf{C}_n$ must exist as well. However, we can have simpler cases of conceptual symmetry also when such a functor T_e does not exist.

Thus, this concept of conceptual symmetry will be introduced for the categories, provided by (partial) composition '$*$' of its objects as well, by the following hierarchy of its properties, from the basic (minimal) to more complete features:

1. Perfectly symmetric categories (with duality operator $B_\top$).
2. Conceptually closed categories (CoCC) with a functor T_e.
3. Symmetry-extended categories.
4. Imploded categories.

The imploded categories, as we will see next, may be considered as the upper bound symmetry case of the n-dimensional levels, and represent the collapse of all n-dimensional levels into their basic category. If we consider, e. g., the $(n+1)$-dimensional level as the metatheory of the n-dimensional level, then in the imploded category we can express all of its metatheories.

Conceptual (internal) symmetry is *local* for basic and perfectly symmetric categories, i. e., it is valid for a given n-dimensional level $\mathbf{C}$ but not necessarily for all other n-dimensional levels, while for more complete features it is global (as the symmetries of the comma-propagation transformation presented in the previous chapter). Next, we will explain some of these introduced concepts by an example of categorial model applied to predicate logic.

Category for predicate logic with intensional abstraction

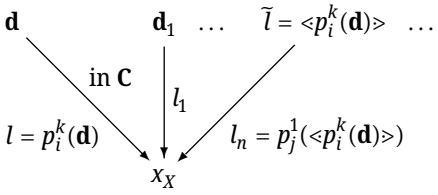
Let us consider, e. g., the categories as models for the predicate logics, where the arrows are the logic sentences and the objects are the terms of the logic. So, for a k-ary predicate letter p_i^k we can represent a ground atom (an atomic sentence) $p_i^k(\mathbf{d})$, where $\mathbf{d} = (d_1, \ldots, d_k)$ is a tuple-term of this logic, as an arrow in this $\mathbf{D}$ category $p_i^k : \mathbf{d} \to x_X$ where x_X is a typed logic-variable (which is a term of the logic) with domain of values X of a given lattice of truth-values $(X, \leq)$. So, this arrow in $\mathbf{D}$ is represented by the object $\langle \mathbf{d}, x_X, p_i^k \rangle$ in the comma (arrow) category $\mathbf{D} \downarrow \mathbf{D}$.

So, if the represented logic by this category $\mathbf{D}$ is the standard First-order Logic (FOL), then we cannot have the functor T_e such that $\widetilde{p_i^k} = T_e(\langle \mathbf{d}, x_X, p_i^k \rangle)$ is a *term*, i. e., an object in this FOL category $\mathbf{D}$, because such a term would be obtained from the logic sentence, and the FOL has not such features (reification).

However, if we now extend the FOL [7] by the unary intensional abstraction operator "<>," which transforms a given sentence ϕ into the new term "<ϕ>," traduced by "that ϕ," we are able to define also the metasentences (sentences about other sentences)

1 Especially, if $f = f_2 \circ f_2$ is a composed arrow in $\mathbf{C}$, how the "conceptualized" object $\widetilde{f_2 \circ f_2}$ is structurally composed by its "conceptualized" components $\widetilde{f_1}$ and $\widetilde{f_2}$.

by using this abstracted term inside other predicate letters of this logic language as, e. g., $p_j^1(<\phi>)$, which express the new metasentence, "I believe that ϕ" where p_j^1 is the unary belief predicate letter. Thus, the Herbrand base of this extended predicate logic with intensional abstraction operator, can be represented by the arrows of the following graph G_H:



(3.2)

The mathematical categories having such a symmetry property are able to represent not only a particular theory but also the metatheories about this basic "initial" particular theory inside *the same* category. So, such categories are similar to the natural language properties, differently from, e. g., the standard (extensional) first-order logic with Tarskian truth semantics [7]. Such properties become important in the area of the Semantic WEB, a proposal to build an infrastructure of machine-readable semantics for the data (ontology) on the World Wide Web. The enormous amount of data has made it difficult to find, access, present and maintain the information required by users. This is because the information content is presented primarily in natural language, and it needs the formal language specification with natural language features, as, e. g., *reification*. In the semantic Web-languages RDF and Web Ontology Language (OWL), we have that the statements are also resources, so that the statement can be applied recursively to statements, allowing their nesting (reification).

In reflective languages, reification data is causally connected to the related reified aspect such that a modification to one of them affects the other. Therefore, the reification data is always a faithful representation of the related reified aspect. *Reification data* is often said to be made a *first class object*. In programming language design, a first-class citizen (also type, object, entity or value) in a given programming language is an entity, which supports all of the operations generally available to other entities. These operations typically include being passed as an argument, returned from a function, modified and assigned to a variable. The concept of first- and second-class objects was introduced by Christopher Strachey in the 1960s when he contrasted real numbers (first-class) and procedures (second-class) in ALGOL. In FOL, we have the variables as arguments inside the predicates and terms, which can be assigned to variables are first-class objects while the predicates are the second-class objects. When we transform a virtual predicate into a term, by using the intensional abstraction operator, we transform a logic formula into the first-class object to be used inside another predicate as first-class objects. Thus, abstracted terms in the intensional FOL [7] are just such abstracted terms as reification of logic formulae. For example, the sentence "Marco thinks *that Zoran runs*," expressed by

thinks(*Marco*, ⩽*runs*(*Zoran*)⩾) by using binary predicate "*thinks*" and unary predicate "*runs*" where the ground atom *runs*(*Zoran*) is reified into the predicate "*thinks*."

Now, if we consider such statements as the morphisms in a given category, the necessity to use such morphisms also as the objects of this category in order to make another statement about them (reification), then we need the categorial symmetry inside this category. This duality "sentences-terms" is an analogy to the duality "morphisms-objects" in the mathematical concept of category, so in this case we will generate a particular symmetry in the category for the Herbrand base, where the morphisms represent the ground atoms in the Herbrand base and the objects are the terms.

So, the intensional abstraction of an atom into a term is just the transformation of a morphism into an object, used in definition of categorial symmetry.

Definition 10. Let us define the graph G_H of the Herbrand base H of the intensional FOL with abstraction operator:

Each atom $p_i^k(\mathbf{d}) \in H$ is represented as an arrow $l : \mathbf{d} \to x_X$ in G_H (left-hand side in diagram (3.2)) from domain $\mathbf{d} = (d_1, \ldots, d_k)$, tuple of values in $\mathcal{D}$, into the logic-variable x_X of the lattice $(X, \leq)$ of the truth values.

We denote by $\mathbf{Sch}(G_H)$ the sketch category derived from this graph G_H where to each object (node) $\mathbf{d}$ is associated the identity arrow equal to the ground atom $\doteq (\mathbf{d}, \mathbf{d})$, and to each object equal to abstracted term ⩽A⩾ for some atom $A \in H$ is associated with the identity arrow equal to the ground atom $\doteq$ (⩽A⩾, ⩽A⩾), where '$\doteq$' is the binary identity predicate. The composition of morphisms (arrows) in $\mathbf{Sch}(G_H)$ is defined by logic conjunction $\wedge$, so that two arrows in it are equal if they are logically equivalent.

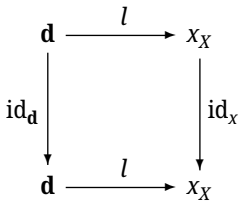
Note that there is not any arrow from the object x_X (which *is not terminal*) toward other objects, and that all identity arrows in category $\mathbf{Sch}(G_H)$ are tautologies, and that the composition of arrows is a logic conjunction, so we can omit the identity arrows from the compositions. The only nonidentity arrows in $\mathbf{Sch}(G_H)$ are that obtained from the ground atoms in H.

If we extend the graph with the morphisms represented by sentences $\phi(\mathbf{x})/g : g(\mathbf{x}) \to x_X$, where $\phi(\mathbf{x})$ is a logic formula with variables in the tuple $\mathbf{x}$, and g is an assignment to the variables, then we obtain the sketch category with categorial symmetry extended to all sentences of the logic language with objects (terms) ⩽$\phi(\mathbf{x})/g$⩾ as well.

Corollary 5. *We have categorial symmetry of morphisms* Mor *and objects* Ob *in* $\mathbf{Sch}(G_H)$.

Proof. Note that in the arrow category $\mathbf{C} \downarrow \mathbf{C}$ where $\mathbf{C} = \mathbf{Sch}(G_H)$, all arrows are just the identity arrows $(\mathrm{id}_{\mathbf{d}}, \mathrm{id}_x) : \langle \mathbf{d}, x_X, l\rangle \to \langle \mathbf{d}, x_X, l\rangle$ or the couples of the same arrow in $\mathbf{C}$, $(l; l) : \langle \mathbf{d}, \mathbf{d}, \mathrm{id}_{\mathbf{d}}\rangle \to \langle x_X, x_X, \mathrm{id}_x\rangle$. So, we can define the functor $T_e = (T_e^0, T_e^1) : (\mathbf{C} \downarrow \mathbf{C}) \longrightarrow \mathbf{C}$ where the arrow component T_e^1 maps the identity arrows in $\mathbf{C} \downarrow \mathbf{C}$ into identity arrows in $\mathbf{C}$, and $T_e^1(l; l) \triangleq l$ is just the arrow in $\mathbf{C}$ from $\mathbf{d}$ to x_X. It is easy to verify that any nonidentity arrow in $\mathbf{C} \downarrow \mathbf{C}$ can be composed (by both sides) only by the identity arrows, so that the functorial property holds also for composed arrows in $\mathbf{C} \downarrow \mathbf{C}$. This is a consequence of the fact that each commutative diagram in $\mathbf{C}$, from diagram (3.2), is

composed by the identity arrows of the same object in **C**, or the following commutative diagrams (for each nonidentity arrow $l : \mathbf{d} \to x_X$ in **C**):



For each object $J(l) = \langle \mathbf{d}, x_X, l \rangle$ in $\mathbf{C} \downarrow \mathbf{C}$, where the ground atom $l = p_i^k(\mathbf{d})$ is a (nonidentity) arrow in **C**, we define that $T_e^0(\langle \mathbf{d}, x_X, l \rangle) \triangleq <l> = <p_i^k(\mathbf{d})>$ is just an abstracted *term*, that is an object in **C**. For the objects obtained from the identity arrows in C, we define that $T_e^0(J(\mathrm{id}_{\mathbf{d}})) = T_e^0(\langle \mathbf{d}, \mathbf{d}, \mathrm{id}_{\mathbf{d}} \rangle) \triangleq \mathbf{d}$ and $T_e^0(J(\mathrm{id}_x)) = T_e^0(\langle x_X, x_X, \mathrm{id}_x \rangle) \triangleq x_X$, which are two objects in **C**.

Consequently, the functor T_e is well-defined and generates (for the non-identity arrows in **C**) the abstracted terms (objects) in **C**. So, we obtained that for each arrow in **C** we have a corresponding object in **C**, and hence the categorial symmetry. □

This presentation was a simple example of how we are using n-dimensional levels (of the arrow categories) in order to obtain a mathematical frame for the definition of the conceptual categorial symmetry (in the sense that this internal symmetry is defined for the two primitive concepts constituting each category: objects and arrows).

Example 7 (Constructive interpretation of logic implication). The *constructive interpretation* of logic implication is different from the classic interpretation with functorial semantics given by $\Rightarrow$: $\mathbf{C}^{\mathrm{OP}} \times \mathbf{C} \to \mathbf{C}$ (covariant for the second and contravariant for the first argument), such that for two objects (logic formulae) a and b, $\Rightarrow^0 (a, b)$ denotes the classic 2-valued implication $a \Rightarrow b$.

In fact, for this constructive interpretation in the *deductive category* **C** where the arrows are well-defined deductions and objects in $C_L(\Gamma)$ are true formulae of a given theory Γ and all other formulae deduced from it (denoted by a closure deduction operation C_L), if we have such a valid deduction of a formula (object) b from a formula a, that is an arrow $f : a \to b$ of **C** than in this category the nonclassic (intuitionistic, for example) implication $a \Rightarrow b$ will be conceptualized object of **C**, obtained from this arrow f. This example, for the natural deduction category **ND**, will be considered in detail in the next chapter.

We will provide other significant examples of categories leaving categorial symmetry for the next chapters of this book. For one of the most complex cases of **DB** category, provided in [5] for the application in big data integration theory based on logic mappings between different database schemas, the Chapter 7 will be dedicated.

3.2 Internal symmetry and metacategory

In what follows, I will use the notion *metacategory* as it was defined in [12], i. e., by a metagraph consisting by objects $a, b, c, \ldots$, arrows (or morphisms) $f, g, h \ldots$ and two operations, *domain*, which assigns to each arrow f an object $a = \mathrm{dom}(f)$, and *codomain* which assigns to each arrow f an object $b = \mathrm{cod}(f)$.

Hence, a *metacategory* is a metagraph with two additional operations:

1. *Identity* (or id), which assigns to each object a an arrow $\mathrm{id}(a)$, with

$$\mathrm{dom}(\mathrm{id}(a)) = \mathrm{cod}(\mathrm{id}(a)) = a$$

   denoted by $\mathrm{id}_a : a \to a$ (i. e., $a \xrightarrow{\mathrm{id}_a} a$).
2. *Composition*, which assigns to each pair of arrows (f, g) with $\mathrm{dom}(g) = \mathrm{cod}(f)$ an arrow $g \circ f$ called *composite*.

These operations in a metacategory are subject to the following axioms:

1. *Associativity* of composition, $k \circ (g \circ f) = (k \circ g) \circ f$, for each well-defined composition of three arrows, i. e., an axiom in FOL:

$$(\forall f)(\forall g)(\forall k)(((\mathrm{cod}(f) = \mathrm{dom}(g)) \wedge (\mathrm{cod}(g) = \mathrm{dom}(k))) \Rightarrow (k \circ (g \circ f) = (k \circ g) \circ f)). \tag{3.3}$$

2. *Unit law*: for each arrow $f : a \to b$ and $g : b \to c$ composition with the identity arrow id_b gives $\mathrm{id}_b \circ f = f$ and $g \circ \mathrm{id}_b = g$, i. e., an axiom in FOL:

$$(\forall f)(\forall g)((\mathrm{cod}(f) = \mathrm{dom}(g)) \Rightarrow ((\mathrm{id}(\mathrm{cod}(f)) \circ f = f) \wedge (g \circ \mathrm{id}(\mathrm{dom}(g)) = g))). \tag{3.4}$$

That is, this notion of a metacategory is used for a category described directly by means of axioms without using the theory of sets. A category (as distinguished from a metacategory) is to be any interpretation of these axioms within set theory. Commutative diagrams in a given category **C** are expressed algebraically by a number of equations, such that

- A *normalized* commutative diagram in **C** is obtained by elimination from it all identity arrows.
- An *atomic* commutative diagram in **C** is expressed algebraically by a single equation. Thus, it can be expressed by a single arrow in the arrow category $\mathbf{C} \downarrow \mathbf{C}$.

Remark. Note that also in the definition of a metacategory, the unique operation for objects is the *Identity* operation, which simply tells us the existence of the identity arrow for each object of a given category. So, this operation does not tell anything *about the objects* but from the fact that it is an operation on objects, we specify only that for a given object *there is a unique* identity arrow of this object. *Identity* operator is necessary only for definition of the unit law in the axiom (3.4).

Unit law in the case when $f = g = \mathrm{id}_b : b \to b$ gives $\mathrm{id}_b \circ \mathrm{id}_b = \mathrm{id}_b$ for each identity arrow, so that for any object b the set of all arrows from b into b is a monoid with its unit equal to this identity arrow id_b. Again, the identity arrow does not tell us anything about the object from which is derived by operation *Identity*, but tells us that it can be eliminated from the composition of the arrows (e. g., in order to obtain the *normalized* commutative diagrams). □

However, if we only represent the objects by their identity arrows, technically in a given category we can work *only by using the arrows*. Generally, in a given category it does not hold for the objects: we cannot represent all arrows by particular objects in a category in order to be able to work technically only with objects. In order to be able to substitute the arrows equivalently by the objects, we must define also the new binary operation '$*$' for composition of the objects, analogous to the operation of composition of the arrows defined in point 2 above. For the categories where such binary operation can be defined as well, we would be able to work technically only by using the arrows, or only by using the objects. In such case of the fundamental symmetry, the primitive categorial concept's objects and arrows are dual, and if we use the symbol $\top$ for any of these two typed primitive concepts, the dual concept will be denoted by the symbol $\bot$.

This internal symmetry between two primitive concepts of the metacategory, objects and arrows, for the symmetry-hierarchy given in previous section, can be formalized by the following definition.

Definition 11. A metacategory, provided by a new operation '$*$' for composition of the objects, is *conceptually symmetric* if it can be described *technically* by each type of the primitive concepts of the metacategory: the objects or the arrows.[2]

That is, for each *atomic normalized* commutative diagram represented by a single equation e_A, there is also the equation e_O (which cannot be expressed in the category because the left- and right-hand side of the equation would be just the same object of the category), obtained by replacement of each arrow in e_A by the corresponding object and the operation '$\circ$' for composition of the arrows by operation '$*$' for the composition of these objects.

The data for an *objects-only metacategory* consist of objects, certain ordered pairs (a, b), called the composable pairs of objects, and the operation '$*$' assigned to each composable pair (a, b) an object $a * b$, called their composite.

Each conceptually symmetric category will be called the *symmetric category*, as the most general kind of the metacategorial symmetry. In what follows, we will develop a

2 While the equations in a category can be represented by commutative diagrams composed by the arrows of the category, the corresponding equations represented by a composition of the objects cannot be represented in the category because both objects on left and right side of an equation would be represented by the same unique object in the category with its identity arrow. So, the equations of composition of the objects has to be supported in a different and noncategorial framework, as it is a standard for the mathematical system of equations.

hierarchy of more specific kinds of categorial symmetries inside this most general kind introduced in Definition 11.

The direct consequence from this definition is that also the operation '$*$' of the composition of the objects of any conceptually symmetric category must be associative as well.

From the fact that any property in a category is expressed by complex commutative diagrams (algebraically representable by a number of equations e_A of atomic normalized subdiagrams) can be equivalently expressed by the same number of equations e_O of corresponding composition of the objects. This is the meaning of the conceptual symmetry between these two primitive types of concepts of a given category.

The banal example for the conceptually symmetric category is any discrete category. But in discrete categories, we have no any atomic normalized commutative diagram. Let us consider a simple example for a category leaving the atomic normalized commutative diagrams with necessity to use the operation '$*$' as well.

Example 8. Let us consider a given preorder category **C**, with arrows $a \xrightarrow{\leq} b$ corresponding to the preorder relationship $a \leq b$ between two objects in **C**.

Then we can enlarge this category **C** by the new objects (having only the identity arrow from and into such an object), into the category **E**, as follows:

Foe each arrow $a \xrightarrow{\leq} b$ in **C**, we add the new object as the ordered pair (a, b), with its identity arrow, into **E**.

Note that by such an extension, the category **C** becomes a subcategory of **E**, that this new extended category **E** remains a preorder category as well, but we have no any normalized commutative diagram composed by the objects in **E**, which are not the objects of its subcategory **C**. That is, each normalized commutative diagram in **E** is also a commutative diagram in **C**, and hence *all* properties present in original category **C** are present (and only these properties) in the extended preorder category **E** as well.

So, for any composition (equation (e_A)) $h = g \circ f$ of two arrow $f : a \xrightarrow{\leq} b$ and $g : b \xrightarrow{\leq} c$, we obtain the composition $(a, b) * (b, c)$ of two objects (a, b) and (b, c) in **E** (which are not present in **C**) so that we obtain the new equation e_O,

$$(b, c) * (a, b) \triangleq (a, b) \cdot (b, c) = (a, c)$$

where the new composition operation for the objects '$*$' in **E** is derived from the associative operation $\cdot$ of composition of binary relations (by exchanging the order of arguments of this operation). Thus, also operation '$*$' for composition of the objects satisfies the associativity of composition, i. e.,

$$\begin{aligned}(c, d) * \big((b, c) * (a, b)\big) &= (c, d) * \big((a, b) \cdot (b, c)\big) \\ &= \big((a, b) \cdot (b, c)\big) \cdot (c, d) \\ &= (a, b) \cdot \big((b, c) \cdot (c, d)\big) \quad \text{from the associativity} \\ &\quad \text{of composition of binary relations}\end{aligned}$$

$$= \big((b,c)\cdot(c,d)\big) * (a,b)$$
$$= \big((c,d) * (b,c)\big) * (a,b).$$

Note that this obtained object (a,c) in **E** is inserted into the category **E** just from the arrow $h : a \xrightarrow{\leq} c$ of the original category **C**.

So, it is easy to verify that from Definition 11, the preorder category **E** is provided by the well-defined (partial) operation "$*$" of objects.

However, the preorder category **E** in the example above is not *perfectly* symmetric, because not for all arrows in **E** we have the corresponding objects: in fact, for each identity arrow of an new object (ordered pair) in **E**, which is not an object in **C**, we have no corresponding object in **E**. So, in what follows, we will define this new more specific *perfect* categorial symmetry.

Definition 12. A category **C** is *perfectly* symmetric if there exists a duality operator $B_{\top}$ from arrows into the objects (which is dual type of categorial primitive concepts), and hence if there exist the composition $g \circ f$ of two arrows in **C**, then we have the following homomorphism:

$$B_{\top}(g \circ f) = B_{\top}(g) * B_{\top}(f) \tag{3.5}$$

where $B_{\top}(g)$ and $B_{\top}(f)$ are the "conceptualized objects" in **C** denoted shortly by $\widetilde{g}$ and $\widetilde{f}$, respectively. The duality operator $B_{\top}$ must satisfy the "representability principle" that for each identity arrow $\mathrm{id}_a : a \to a$ it must be valid for the following isomorphism in **C**:

$$B_{\top}(\mathrm{id}_a) \simeq a \tag{3.6}$$

corresponding to the basic categorial *Identity* operation id.

The objects obtained from arrows will be called *conceptualized* objects. For example, let us consider the polynomial function $f(x) = (x^2 - 3x + 1)$, with the domain being the closed interval of reals from 0 to 1, and codomain $\mathbb{R}$ of all reals. It is an arrow $f : [0,1] \to \mathbb{R}$ in the category **Set**. The conceptualized object obtained from this arrow, denoted by $\widetilde{f}$, is the *set* (hence an object in **Set**) equal to the graph of this polynomial function. That is, $\widetilde{f} = \{(x, f(x)) \mid x \in [0,1]\} \in \mathrm{Ob}_{\mathrm{Set}}$.

Note that in this definition we do not need the bijection for the operator $B_{\top}$, as we can show in next example.

It is easy to verify that the property (3.5) cannot be satisfied for the case of the conceptual symmetry in Example 8. If $g = f = \mathrm{id}_a$ is equal to the identity arrow of an object a of **E**, which is not an object of **C**, the object $B_{\top}(\mathrm{id}_a)$ does not exist in the extended category **E** as well (differently from representability principle in (3.6)). However, let us show that the **Set** category is perfectly symmetric.

Example 9. Let us consider two arrows that are not surjective functions in **Set**, $f : a \to b$ and $g : b \to c$, with their composite $h = g \circ f : a \to c$. We will denote by $\mathrm{im}(f)$ the image of the function f, that is, the set $\mathrm{im}(f) = \{f(x) \mid x \in a\}$, so that in **Set** there are also the following surjective functions $f' : a \twoheadrightarrow \mathrm{im}(f)$, $g' : b \twoheadrightarrow \mathrm{im}(g)$ and $h' : a \twoheadrightarrow \mathrm{im}(g \circ f)$, which are equal to f, g and h respectively but only have the strict subsets of their codomains, respectively.

So, we can define B_{T} for each arrow k of **Set** by

$$\widetilde{k} = B_{\mathsf{T}}(k) \triangleq \{(x, k(x)) \mid x \in \mathrm{dom}(k)\} \tag{3.7}$$

i. e., the obtained object $B_{\mathsf{T}}(k)$ is the graph of the function k. Consequently, we obtain that for the arrows $f \neq f' : a \to \mathrm{im}(f)$, $g \neq g' : b \to \mathrm{im}(g)$ and $h \neq h' : a \to \mathrm{im}(h)$, so that f', g' and h' are surjective reductions of the f, g and h, relatively,

$$B_{\mathsf{T}}(f) = B_{\mathsf{T}}(f'), \quad B_{\mathsf{T}}(g) = B_{\mathsf{T}}(g') \quad \text{and} \quad B_{\mathsf{T}}(h) = B_{\mathsf{T}}(h'), \tag{3.8}$$

and hence, the operation B_{T} cannot be generally bijective (have no inverse).

Note that in this case, from (3.5), for the composition $h = g \circ f$ we obtain that

$$\begin{aligned}
\widetilde{h} &= B_{\mathsf{T}}(h) \\
&= B_{\mathsf{T}}(g \circ f) \\
&= B_{\mathsf{T}}(g) * B_{\mathsf{T}}(f) \\
&= \{(y, g(y)) \mid y \in b\} * \{(x, f(x)) \mid x \in a\} \\
&= \{(x, g(y)) \mid x \in a \text{ and } y = f(x)\} \\
&= \{(x, g(f(x))) \mid x \in a\} \\
&= \{(x, (g \circ f)(x)) \mid x \in a\} \\
&= \{(x, h(x)) \mid x \in a\}
\end{aligned}$$

as expected, and '$*$' is derived from composition of binary relations '$\cdot$' by

$$\widetilde{g} * \widetilde{f} = B_{\mathsf{T}}(g) * B_{\mathsf{T}}(f) = B_{\mathsf{T}}(f) \cdot B_{\mathsf{T}}(g) = \widetilde{h}$$

and hence, we have for a given equation $h = g \circ f$, the following transformation of equations of composition of the arrows into the dual equation of composition of their conceptualized objects:

$$h = g \circ f \quad \text{implies} \quad \widetilde{h} = \widetilde{g} * \widetilde{f}. \tag{3.9}$$

Consequently, any commutative diagram, composed by a number of atomic diagrams (equations), can be represented by the same number of equations between the conceptualized objects obtained from the arrows in this commutative diagram.

Another example of perfectly symmetric category, considered in detail in the next chapter, is the natural deduction category **ND** for constructive interpretation of logic implication, introduced previously in Example 7 at the end of Section 3.1.

As we will see next, such a property (3.5) is valid for the conceptually closed categories, and symmetry-extended categories and imploded categories as well. The last case in the bottom of this hierarchy of symmetrical categories is interesting because represents a limit case for *n*-dimensional levels presented in previous chapters, and essentially exhibits the nature of the relationship between the internal (conceptual) symmetry and infinite structure of *n*-dimensional levels.

Proposition 6. *Let* $\mathbf{C}$ *be perfectly symmetric category with duality operator* $B_\top$ *satisfying the properties* (3.5) *and* (3.6)*. Then we define the "conceptual lifting" operator, which transforms* $\mathbf{C}$ *into category of conceptualized objects* $\widetilde{\mathbf{C}}$*, such that:*

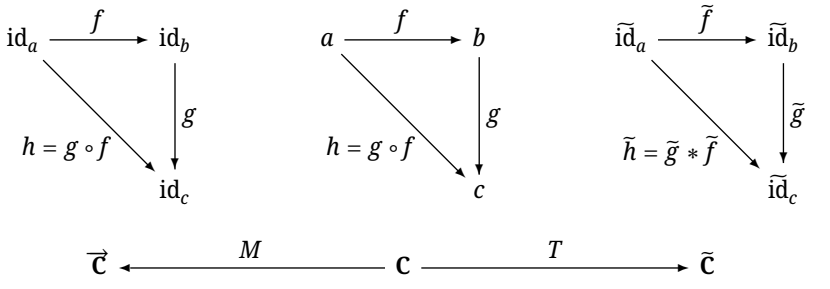
1. *Each arrow in* $\widetilde{\mathbf{C}}$ *is a conceptualized object* $\widetilde{f} = B_T(f)$ *for* $f \in \mathrm{Mor}_{\mathbf{C}}$.
2. *Each object in* $\widetilde{\mathbf{C}}$ *is a conceptualized object* $\widetilde{\mathrm{id}_a} = B_T(\mathrm{id}_a)$ *for identity arrow* $\mathrm{id}_a \in \mathrm{Mor}_{\mathbf{C}}$*, and hence* $\mathrm{Ob}_{\widetilde{\mathbf{C}}} \subset \mathrm{Mor}_{\widetilde{\mathbf{C}}}$.

Thus, there is a transformation functor $T = (T^0, T^1) : \mathbf{C} \to \widetilde{\mathbf{C}}$ *with the components*

$$T^0 \triangleq B_\top \tau_I, \quad T^1 \triangleq B_\top \tag{3.10}$$

where $\tau_I : \mathrm{Id}_{\mathbf{C}} \xrightarrow{\cdot} \mathrm{Id}_{\mathbf{C}}$ *is the identity natural transformation for the category* $\mathbf{C}$.

Proof. It is well known that each category $\mathbf{C}$ can be equivalently replaced by the category $\overrightarrow{\mathbf{C}}$ composed by the arrows only in which the objects are represented by their identity arrows, and hence with the functor $M = (\tau_I, \mathrm{id}) : \mathbf{C} \to \overrightarrow{\mathbf{C}}$, where the component of this functor for the arrows is the identity function. So, if $\mathbf{C}$ is perfectly symmetric as well, then we can define also the category $\widetilde{\mathbf{C}}$ composed by only conceptualized objects $\widetilde{f} = B_\top(f)$ for each arrow f in $\mathbf{C}$ as specified in this proposition, and with the duality functor $D = (B_\top, B_\top) : \overrightarrow{\mathbf{C}} \to \widetilde{\mathbf{C}}$, and hence by composition we obtain the functor $T = D \circ M$. Also, this passive transformation T of "conceptual lifting" from $\mathbf{C}$ into $\widetilde{\mathbf{C}}$ is an invariance w. r. t. the base category $\mathbf{C}$, i. e., represents the perfect categorial symmetry of three "reference frames of Galilean boost," $\mathbf{C}$, $\overrightarrow{\mathbf{C}}$ and $\widetilde{\mathbf{C}}$ (see more in Section 2.5):



(3.11)

Note that differently from the comma-propagation passive "Galilean boost" transformations, which there exist always, the passive conceptual-lifting transformation there exists only for the perfectly symmetric categories satisfying Definition 12. □

The phenomena of internal symmetry

Based on the correspondence with physics theory provided in Section 2.5, where each (nonfunctor) category is considered as a reference inertial system, i. e., a flat space with points corresponding to objects of a category and arrows as oriented paths between points of this space, the conceptually symmetric category **C** and derived from it $\overrightarrow{\mathbf{C}}$ and $\widetilde{\mathbf{C}}$ can be considered as three different reference systems.

In our case of a *unique* internal symmetry, we have a unique field Ψ and unique transformation from arrows to the (conceptualized) objects in **C**. We can follow the correspondence provided in Section 2.5, where for the comma-propagation transformations each Ψ is a *adjunction-as-field*, i. e., *the set* of all commutative diagrams (2.52) for every point p_i in **C** and diagram d_i', and hence a *subset of equations* corresponding to these commutative diagrams of an adjunction. Consequently, for a given *perfectly symmetric category* **C** this unique field Ψ is just the complete set of equations representing *all commutative diagrams* in **C**, with restriction that each arrow in these diagrams is not a composition of the same identity arrow (the composition of an identity arrow by itself is redundant).

So, the Lagrangian density $\mathcal{L}(\Psi, p_i)$, for a given *perfectly symmetric category* **C**, in a point p_i is analogously (w. r. t. the comma-propagation transformations) defined as the set of all arrows f with its domain $\mathrm{dom}(f) = p_i$, such that are components of some universal arrow in **C**.

Consequently, based on (3.9), any commutative diagram in a *perfectly symmetric category* **C**, composed by a number of atomic diagrams (equations), can be represented by the same number of equations between the conceptualized objects obtained from the arrows in this commutative diagram (see the diagrams in (3.11)). Thus, the *transformation* of the arrows into their conceptualized objects in **C** does not modify the mathematical form of the (total) field Ψ and its Lagrangian density $\mathcal{L}(\Psi, p_i)$, and hence from this invariance we obtain this phenomena of categorial *internal symmetry*.

3.3 Conceptually closed categories: A topology

The concept of a topology, conceptually closed categories, has been introduced in my PhD thesis [4] with a number of its applications. This kind of more specific category symmetry is obtainable when the duality operator B_{T} introduced for the conceptually symmetric categories in Definition 12, can be extended to a functor. That is, by using standard categorial language (because of that we call this symmetry "*closed* by functor"),

as it was anticipated by scheme (3.1) in the introduction of Section 3.1, with a descending functor $T_e = (T_e^0, T_e^1) : \mathbf{C} \downarrow \mathbf{C} \to \mathbf{C}$, such that $B_\top = T_e^0 J$.

It has to be "inverse" (up to isomorphism) functor in **Cat** to the diagonal functor $\blacktriangle = (\blacktriangle^0, \blacktriangle^1) : \mathbf{C} \longrightarrow (\mathbf{C} \downarrow \mathbf{C})$ introduced in (1.11), so that the natural transformations φ and φ^{-1}, defined by (1.23) in Example 3 in Section 1.3, become two natural isomorphisms. We recall that, for any object a in a category $\mathbf{C}$, $\blacktriangle^0(a) = \langle a, a, \mathrm{id}_a \rangle = J(\mathrm{id}_a)$, and for any arrow f, $\blacktriangle^1(f) = (f; f)$.

An important subset of symmetric categories are the Conceptually Closed Categories (CoCC), specified as follows.

Proposition 7. *A conceptually closed category (CoCC) is a* perfectly *symmetric category* $\mathbf{C}$ *for which there exists a functor* $T_e = (T_e^0, T_e^1) : (\mathbf{C} \downarrow \mathbf{C}) \longrightarrow \mathbf{C}$, *"inverse" to diagonal functor* $\blacktriangle$, *such that for the object-component of this functor* $T_e^0 = B_T \psi$, *where* $\psi : F_{\mathrm{st}} \stackrel{\cdot}{\longrightarrow} S_{\mathrm{nd}}$ *is the natural transformation with associated function* (1.2), $J^{-1} : \mathrm{Ob}_{\mathbf{C}_2} \to \mathrm{Mor}_{\mathbf{C}}$ *inverse to the encapsulation operator* J.

The arrow-component of this closure functor has to satisfy, for each arrow $(f; f) : J(\mathrm{id}_{\mathrm{dom}(f)}) \to J(\mathrm{id}_{\mathrm{dom}(f)})$,

$$T_e^1(f; f) = \varphi^{-1}(\mathrm{cod}(f)) \circ f \circ \varphi(\mathrm{dom}(f)) \tag{3.12}$$

where the natural transformations φ *and* φ^{-1} *in* (1.23) *are just two natural* isomorphisms

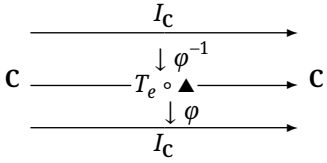
$$\varphi : T_e \circ \blacktriangle \simeq I_{\mathbf{C}} \quad \textit{and} \quad \varphi^{-1} : I_{\mathbf{C}} \simeq T_e \circ \blacktriangle \tag{3.13}$$

where $I_{\mathbf{C}}$ *is the identity endofunctor for* $\mathbf{C}$.

Proof. For any object $J(f)$ in the arrow category $\mathbf{C}_2 = \mathbf{C} \downarrow \mathbf{C}$, we have that $T_e(J(f)) = B_\top \psi(J(f)) = B_\top J^{-1} J(f) = B_\top(f)$ is the conceptualized object $\bar{f}$ in $\mathbf{C}$. That is $B_T = T_e^0 J$, and from the (3.6), for each identity arrow $\mathrm{id}_a : a \to a$ in $\mathbf{C}$ must hold the isomorphism $B_\top(\mathrm{id}_a) \simeq a$, and hence there must exists also the natural isomorphism $\varphi : T_e \circ \blacktriangle \simeq I_{\mathbf{C}}$ and $\varphi^{-1} : I_{\mathbf{C}} \simeq T_e \circ \blacktriangle$, where $I_{\mathbf{C}}$ is the identity endofunctor for $\mathbf{C}$.

In fact, for each object a of $\mathbf{C}$, we have that $\blacktriangle(a) = J(\mathrm{id}_a)$ so that $(T_e \circ \blacktriangle)(a) = T_e(J(\mathrm{id}_a)) = B_\top \psi(J(\mathrm{id}_a)) = B_\top(\mathrm{id}_a)$, and hence the arrow component of natural transformation $\varphi(a) : B_\top(\mathrm{id}_a) \to a$ is isomorphic from (3.6).

These natural isomorphisms (3.13) are the specific case of the natural transformations $\varphi : T_e \circ \blacktriangle \stackrel{\cdot}{\longrightarrow} I_{\mathbf{C}}$ and $\varphi^{-1} : I_{\mathbf{C}} \stackrel{\cdot}{\longrightarrow} T_e \circ \blacktriangle$ introduced by (1.23) in Example 3 in Section 1.3 with diagram of vertical compositions (1.24),



for which, in our case of CoCC $\mathbf{C}$, we have that both vertical compositions, $\varphi^{-1} \bullet \varphi : T_e \circ \blacktriangle \xrightarrow{\cdot} T_e \circ \blacktriangle$ and $\varphi \bullet \varphi^{-1} : I_{\mathbf{C}} \xrightarrow{\cdot} I_{\mathbf{C}}$, generate the identity arrows in this conceptually closed category $\mathbf{C}$.

Let us show that (3.12) satisfy the functorial properties of T_e. In fact, for the identity arrow $f = \mathrm{id}_c : c \to c$, we obtain that $T_e(\mathrm{id}_c; \mathrm{id}_c) = \varphi^{-1}(c) \circ \mathrm{id}_c \circ \varphi(c) = (\varphi^{-1} \bullet \varphi)(c)$, which is the identity arrow of the conceptualized object $\widetilde{\mathrm{id}}_c = T_e(J(\mathrm{id}_c))$. For a composed arrow $g \circ f$, we have from the fact that $\mathrm{dom}(g) = \mathrm{cod}(f)$,

$$\begin{aligned} T_e(g \circ f; g \circ f) &= \varphi^{-1}(\mathrm{cod}(g \circ f)) \circ g \circ f \circ \varphi(\mathrm{dom}(g \circ f)) \\ &= \varphi^{-1}(\mathrm{cod}(g)) \circ g \circ f \circ \varphi(\mathrm{dom}(f)) \\ &= \varphi^{-1}(\mathrm{cod}(g)) \circ g \circ (\varphi(\mathrm{dom}(g)) \circ \varphi^{-1}(\mathrm{cod}(f))) \circ f \circ \varphi(\mathrm{dom}(f)) \\ &= T_e(g; g) \circ T_e(f; f). \end{aligned}$$

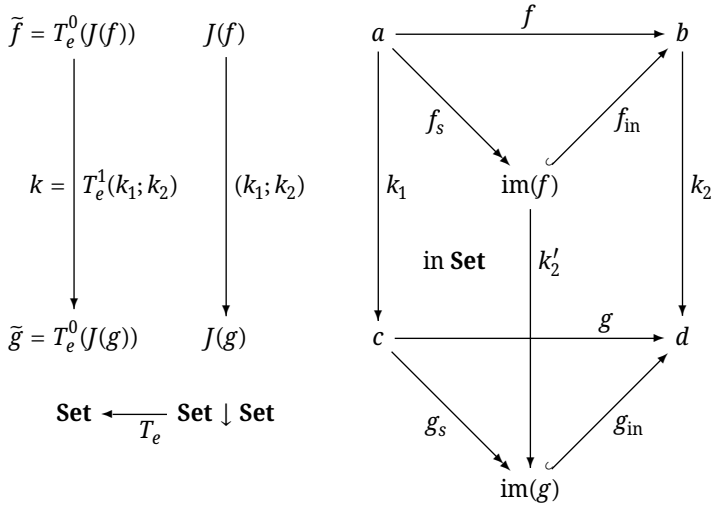
□

So, this natural isomorphism is a reason for the *conceptual functorial closure* of such symmetric categories, existing between the objects and their identity arrows. By this "closure," we mean that the identity arrows do not generate the new objects (up to isomorphism).

Let us consider the following simple example of conceptually closed categories.

Corollary 6. *The category* **Set** *is a Conceptually Closed Category (CoCC).*

Proof. We will continue with work presented in the Example 9, by considering that each function $f : a \to b$ with image $\mathrm{im}(f) \subseteq b$ can be represented as a composition of a surjective function $f_s : \longrightarrow\!\!\!\!\!\rightarrow \mathrm{im}(f)$ and an inclusion $f_{\mathrm{in}} : \mathrm{im}(f) \hookrightarrow b$, so that an arrow $(k_1; k_2) : J(f) \to J(g)$ in **Set** $\downarrow$ **Set** can be represented by the following commutative diagram in **Set**:



where the function $k = T_e(k_1;k_2) : \widetilde{f} \to \widetilde{g}$, we have that of the any element (a pair), $(x,f(x)) \in \widetilde{f}$; this function k is defined by

$$k(x,f(x)) \triangleq (k_1(x),k_2(f(x))) = (k_1(x),(k_2 \circ f)(x)) \in \widetilde{g}. \tag{3.14}$$

Notice that with such surjective-injective decomposition of the functions, we obtain that $\widetilde{f} = T_e(J(f)) = T_e(J(f_s))$ and $T_e(k_1,k_2') : J(f_s) \to J(g_s)$ is the reduction of the function $k = T_e(k_1;k_2)$ to the surjective commutative component $g_s \circ k_1 = k_2' \circ f_s$ of the commutative diagram $g \circ k_1 = k_2 \circ f$ shown in the commutative diagram above. It is easy to show that the functor T_e : **Set** $\downarrow$ **Set** $\to$ **Set** is well-defined:

1. For each identity arrow $(\mathrm{id}_a;\mathrm{id}_b) : J(f) \to J(f)$ in **Set** $\downarrow$ **Set**, the arrow $k = T_e(\mathrm{id}_a;\mathrm{id}_b)$ from Definition (3.14), we obtain that k is the identity function $\mathrm{id}_{\widetilde{f}}$ of the conceptualized object $\widetilde{f} = T_e(J(f))$.
2. For the composition of the arrows $(k_1;k_2) : J(f) \to J(g)$ and $(h_1;h_2) : J(g) \to J(m)$, by using definition in (3.14), it holds the functorial property $T_e(h_1 \circ k_1; h_2 \circ k_2) = T_e((h_1;h_2) \circ (k_1;k_2)) = T_e(h_1;h_2) \circ T_e(k_1;k_2)$. □

We can show that each CoCC introduces two particular natural transformations.

Corollary 7. *Let* **C** *be a conceptual closed category. Then there are two particular natural transformations* $\tau : F_{\mathrm{st}} \xrightarrow{\cdot} T_e$ *and* $\tau^{-1} : T_e \xrightarrow{\cdot} S_{\mathrm{nd}}$*, for the first and second projections* $F_{\mathrm{st}}, S_{\mathrm{nd}}$: **C** $\downarrow$ **C** $\to$ **C** *and this new "middle projection"* T_e *of the conceptual closure, defined by the following vertical composition:*

$$\tau = (T_e\sigma) \bullet (\varphi^{-1}F_{\mathrm{st}}) \quad \textit{and} \quad \tau^{-1} = (\varphi S_{\mathrm{nd}}) \bullet (T_e\sigma^{-1}) \tag{3.15}$$

where the natural transformations $\sigma : \blacktriangle \circ F_{\mathrm{st}} \xrightarrow{\cdot} \mathrm{Id}_{\mathbf{C}_n}$ *and* $\sigma^{-1} : \mathrm{Id}_{\mathbf{C}_n} \xrightarrow{\cdot} \blacktriangle \circ S_{\mathrm{nd}}$ *for* $n = 2$ *are given in Proposition* 2 *by* (1.13)*, and* $\varphi : T_e \circ \blacktriangle \simeq I_C$ *is the natural isomorphism in* (3.13) *with its inverse* φ^{-1}*, and hence*

$$\tau^{-1} \bullet \tau = (\varphi S_{\mathrm{nd}}) \bullet (T_e(\psi;\psi)) \bullet (\varphi^{-1}F_{\mathrm{st}}) \tag{3.16}$$

shown by the following diagram:

$$\begin{array}{ccc} & \xrightarrow{\quad F_{\mathrm{st}} \quad} & \\ & \downarrow \tau & \\ \mathbf{C} \downarrow \mathbf{C} & \xrightarrow{\quad T_e \quad} & \mathbf{C} \\ & \downarrow \tau^{-1} & \\ & \xrightarrow[\quad S_{\mathrm{nd}} \quad]{} & \end{array} \tag{3.17}$$

Proof. We have from (A.10) that $T_e\sigma$ and $T_e\sigma^{-1}$ are well-defined natural transformations, and from (A.9) also $\varphi^{-1}F_{\mathrm{st}}$ and φS_{nd} are well-defined natural transformations. Consequently, the vertical composition of these natural transformations defines the composed natural transformations τ and τ^{-1} as well. So, for each arrow $(h;k) : J(f) \to J(g)$ in

$\mathbf{C}_2 = \mathbf{C} \downarrow \mathbf{C}$, with $f : a \to b$ and $g : c \to d$ two arrows in $\mathbf{C}$, we obtain the following commutative diagram for τ:

$$\begin{array}{ccccccc}
J(f) & & a = F^0_{\mathrm{st}}(J(f)) & \xrightarrow{\varphi^{-1}(a)} & \widetilde{\mathrm{id}_a} = T^0_e\blacktriangle^0(a) & \xrightarrow{T^1_e\sigma(J(f))} & \widetilde{f} = T^0_e(J(f)) \\
\Big\downarrow{\scriptstyle (h;k)} & & \Big\downarrow{\scriptstyle h = F^1_{\mathrm{st}}(h;k)} & & \Big\downarrow{\scriptstyle T^1_e\blacktriangle^1(h)} & & \Big\downarrow{\scriptstyle T_e(h;k)} \\
J(g) & & c = F^0_{\mathrm{st}}(J(g)) & \xrightarrow{\varphi^{-1}(c)} & \widetilde{\mathrm{id}_c} = T^0_e\blacktriangle^0(c) & \xrightarrow{T^1_e\sigma(J(g))} & \widetilde{g} = T^0_e(J(g)) \\
\text{in } \mathbf{C}_2 & & F_{\mathrm{st}} & \xrightarrow{\varphi^{-1}F_{\mathrm{st}}} & T_e\blacktriangle F_{\mathrm{st}} & \xrightarrow{T_e\sigma} & T_e
\end{array}$$

where the commutative diagram in $\mathbf{C}$ is the result of application of the vertical composition of two natural transformations that compose τ as shown in the bottom line of the scheme above. So, the top horizontal arrow of this commutative diagram is equal to $\tau(J(f))$. Thus, for each arrow $f : a \to b$ in $\mathbf{C}$, we have also these two arrows in $\mathbf{C}$ for its conceptualized object $\widetilde{f}$:

$$\tau(J(f)) : a \to \widetilde{f} \quad \text{and, analogously,} \quad \tau^{-1}(J(f)) : \widetilde{f} \to b. \tag{3.18}$$

Equation (3.16) is derived easily as follows:

$$\begin{aligned}
\tau^{-1} \bullet \tau &= ((\varphi S_{\mathrm{nd}}) \bullet (T_e\sigma^{-1})) \bullet ((T_e\sigma) \bullet (\varphi^{-1}F_{\mathrm{st}})) \quad \text{from (3.15)} \\
&= (\varphi S_{\mathrm{nd}}) \bullet ((T_e\sigma^{-1}) \bullet (T_e\sigma)) \bullet (\varphi^{-1}F_{\mathrm{st}}) \quad \text{from associativity of composition } \bullet \\
&= (\varphi S_{\mathrm{nd}}) \bullet (T_e(\sigma^{-1} \bullet \sigma)) \bullet (\varphi^{-1}F_{\mathrm{st}}) \quad \text{from functorial property of } T_e \\
&= (\varphi S_{\mathrm{nd}}) \bullet (T_e(\psi,\psi)) \bullet (\varphi^{-1}F_{\mathrm{st}}) \quad \text{from Proposition 2, } \sigma^{-1} \bullet \sigma = (\psi;\psi). \qquad \square
\end{aligned}$$

It is easy to verify, from the fact that $\sigma = (\psi\blacktriangle^0 F^0_{\mathrm{st}}; \psi)$ and $\sigma^{-1} = (\psi, \psi\blacktriangle^0 S^0_{\mathrm{nd}})$, that in conceptually closed categories the following is valid as well:

$$\tau = (T^1_e(\tau_I F^0_{\mathrm{st}}; \psi)) \bullet (\varphi^{-1}F^0_{\mathrm{st}}), \quad \tau^{-1} = (\varphi^{-1}S^0_{\mathrm{nd}}) \bullet (T^1_e(\psi; \tau_I S^0_{\mathrm{nd}})),$$

where $\tau_I : I_C \longrightarrow I_C$ is an identity natural transformation (for any object a in $\mathbf{C}$, $\tau_I(a) = \mathrm{id}_a$).

Thus, from the diagram in the proof above, we obtain the following commutative diagram:

$$\begin{array}{ccccc}
a = \mathrm{dom}(f) & \xrightarrow{\tau(J(f))} & \widetilde{f} & \xrightarrow{\tau^{-1}(J(f))} & b = \mathrm{cod}(f) \\
\Big\downarrow{\scriptstyle \varphi^{-1}(a)} & & \Big\downarrow{\scriptstyle \mathrm{id}_{\widetilde{f}}} & & \Big\uparrow{\scriptstyle \varphi(b)} \\
\widetilde{\mathrm{id}_a} = T^0_e(J(\mathrm{id}_a)) & \xrightarrow{T^1_e(\mathrm{id}_a; f)} & \widetilde{f} = T^0_e(J(f)) & \xrightarrow{T^1_e(f; \mathrm{id}_b)} & \widetilde{\mathrm{id}_b} = T^0_e(J(\mathrm{id}_b))
\end{array} \tag{3.19}$$

In order to define the class of CoCC, we used the natural isomorphism $\varphi : T_e \circ \blacktriangle \simeq I_C$ in (3.13) with its inverse φ^{-1}, with composition of functors $T_e \circ \blacktriangle \simeq I_C$, so it is natural to investigate the conditions when these two functors are mutually adjoint.

Proposition 8. *Let* $\mathbf{C}$ *be a CoCC closed by functor* $T_e : \mathbf{C}_2 \to \mathbf{C}$*, with* $\mathbf{C}_2 = \mathbf{C} \downarrow \mathbf{C}$*, such that there exists the natural transformation* $\varrho : T_e \xrightarrow{\cdot} F_{\mathrm{st}}$ *as well, and hence the following vertical composition of natural transformations* $\psi \bullet \varrho$*:*

$$\mathbf{C} \downarrow \mathbf{C} \;\underset{\downarrow \psi}{\overset{T_e}{\underset{S_{\mathrm{nd}}}{\overset{\downarrow \varrho}{\underset{}{\overset{}{F_{\mathrm{st}}}}}}}}\; \mathbf{C} \tag{3.20}$$

Then there is the following adjunction $(\blacktriangle, T_e, \varepsilon_e, \eta_e)$ *where the unit is a natural isomorphism* $\eta_e = \varphi^{-1} : I_{\mathbf{C}} \xrightarrow{\cdot} T_e \circ \blacktriangle$ *and counit* $\varepsilon_e : \blacktriangle \circ T_e \xrightarrow{\cdot} I_{\mathbf{C}_2}$ *is defined as a pair of natural transformations,*

$$\varepsilon_e \triangleq (\varrho,\ \psi \bullet \varrho), \tag{3.21}$$

so that we obtain the following identity of natural transformations:

$$T_e \varepsilon_e = \varphi T_e : T_e \circ \blacktriangle \circ T_e \xrightarrow{\cdot} T_e. \tag{3.22}$$

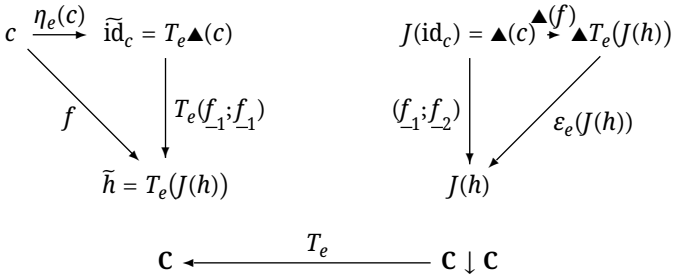
Proof. Thus, for any object $J(h)$ in $\mathbf{C}_2 = \mathbf{C} \downarrow \mathbf{C}$, where $h : a \to b$ is an arrow in $\mathbf{C}$, from the natural transformation ε_e defined by (3.21), we obtain that the arrow component $\varepsilon_e(J(h)) : \blacktriangle \circ T_e(J(h)) \to J(h)$, where $\blacktriangle^0 \circ T_e^0(J(h)) = \blacktriangle^0(\widetilde{h}) = J(\mathrm{id}_{\widetilde{h}})$ and

$$\begin{aligned}
\varepsilon_e(J(h)) &= (\varrho, \psi \bullet \varrho)(J(h)) \\
&= (\varrho(J(h)), (\psi \bullet \varrho)(J(h))) \\
&= (\varrho(J(h)); \psi(J(h)) \circ \varrho(J(h))) \\
&= (\varrho(J(h)); h \circ \varrho(J(h))) : J(\mathrm{id}_{\widetilde{h}}) \to J(h)
\end{aligned}$$

is represented by the following commutative diagram in $\mathbf{C}$:

$$\begin{array}{ccc}
J(\mathrm{id}_{\widetilde{h}}) = \blacktriangle^0 \circ T_e^0(J(h)) & & \widetilde{h} = T_e^0(J(h)) \xrightarrow{\mathrm{id}_{\widetilde{h}}} \widetilde{h} \\
\varepsilon_e(J(h)) \downarrow & \Leftrightarrow & \varrho(J(f)) \downarrow \qquad \text{in } \mathbf{C} \qquad \downarrow h \circ \varrho(J(f)) \\
J(h) & & a = F_{\mathrm{st}}^0(J(h)) \xrightarrow{h} b
\end{array}$$

and hence we have the following adjunction commutative diagram for a given universal arrow $(J(\mathrm{id}_c), \eta(c))$:



(3.23)

where $(\underline{f}_{-1};\underline{f}_{-2}) = \varepsilon_e(J(h)) \circ \blacktriangle(f) = \varepsilon_e(J(h)) \circ (f;f)$, so that

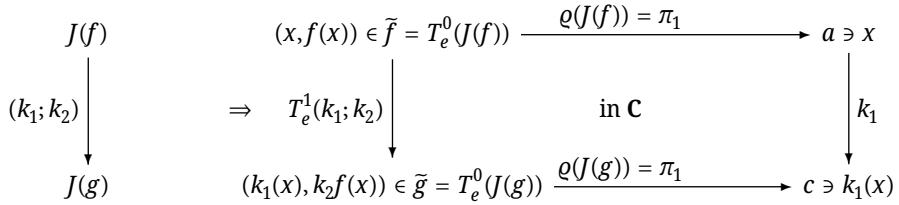
$$\begin{aligned} T_e(\underline{f}_{-1};\underline{f}_{-2}) &= T_e(\varepsilon_e(J(h)) \circ (f;f)) \\ &= T_e(\varepsilon_e(J(h))) \circ T_e(f;f) \quad \text{from the functorial property} \\ &= \varphi T_e(J(h)) \circ T_e(f;f) \quad \text{from (3.22)} \\ &= \varphi(\tilde{h}) \circ T_e(f;f) \\ &= \varphi(\tilde{h}) \circ \varphi^{-1}(\mathrm{cod}(f)) \circ f \circ \varphi(\mathrm{dom}(f)) \quad \text{from (3.12)} \\ &= \varphi(\tilde{h}) \circ \varphi^{-1}(\tilde{h}) \circ f \circ \varphi(c) \quad \text{from the arrow } f : c \to \tilde{h} \end{aligned}$$

i. e., from the identity of composition $\varphi(\tilde{h}) \circ \varphi^{-1}(\tilde{h})$ of an isomorphism with its inverse,

$$T_e(\underline{f}_{-1};\underline{f}_{-2}) = f \circ \varphi(c), \tag{3.24}$$

and hence we obtain the commutativity of the left-hand side of the diagram (3.23), i. e., $T_e(\underline{f}_{-1};\underline{f}_{-2}) \circ \eta_e(c) = (f \circ \varphi(c)) \circ \varphi^{-1}(c) = f \circ \mathrm{id}_c = f$. □

A significant example of the CoCC with adjunction $(\blacktriangle, T_e, \varepsilon_e, \eta_e)$ in Proposition 8 is the category **Set** where each arrow component of natural transformation ϱ is the first projection function $\pi_1 : A \times B \to A$, for $f : a \to b$ and $g : c \to d$ with $g \circ k_1 = k_2 \circ f$ in **C**,



Based on the definition in (3.14), and $k_2 f(x) = g k_1(x)$.

In such conceptually symmetric categories, we can present the conceptualized objects as a particular kind of *limits*.

Corollary 8. *Let* **C** *be a CoCC with adjunction* $(\blacktriangle, T_e, \varepsilon_e, \eta_e)$ *in Proposition* 8. *Then for the small index category* **J** *composed by the single arrow* $l_{12} : a_1 \to a_2$, *there is an adjunction* $(\triangle, G, \varepsilon, \eta_e)$ *of the diagonal functor* $\triangle : \mathbf{C} \to \mathbf{C}^{\mathbf{J}}$ *with functor G such that:*

1. *For each object (functor)* $d' : \mathbf{J} \to \mathbf{C}$ *in* $\mathbf{C}^{\mathbf{J}}$*, the* limit object *in* $\mathbf{C}$ *is*

$$\mathrm{Lim}_{d'} = G^0(d') \triangleq T_e^0(J(d'(l_{12}))) \tag{3.25}$$

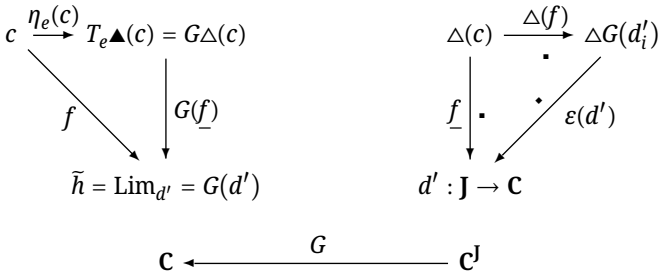
and the counit ε *has the arrow component in* $\mathbf{C}^{\mathbf{J}}$ *(a natural transformation)* $\varepsilon(d')$ *such that*

$$\varepsilon(d')(a_1) \triangleq \varrho(J(d'(l_{12}))) \quad \textit{and} \quad \varepsilon(d')(a_2) \triangleq (\psi \bullet \varrho)(J(d'(l_{12}))). \tag{3.26}$$

2. *For each arrow (natural transformation)* $\underline{f}$ *in* $\mathbf{C}^{\mathbf{J}}$*,*

$$G^1(\underline{f}) \triangleq T_e^1(\underline{f}(a_1); \underline{f}(a_2)). \tag{3.27}$$

Proof. Let us show that for any object c in $\mathbf{C}$ and diagram $d' : \mathbf{J} \to \mathbf{C}$ such that $a = d'(a_1)$, $b = d'(a_2)$ and $d'(l_{12}) = h : a \to b$ is an arrow in $\mathbf{C}$, the following adjunction diagram, with limit object equal to the conceptualized object $\tilde{h}$, commutes:



(3.28)

Let us show that we have the following identity for functor composition:

$$G \circ \triangle = T_e \circ \blacktriangle : \mathbf{C} \to \mathbf{C}. \tag{3.29}$$

In fact, for each object c in $\mathbf{C}$,

$$\begin{aligned} G^0\triangle^0(c) &= T_e^0(J((\triangle^0(c))(l_{12}))) \quad \text{from (3.25)} \\ &= T_e^0(J(\mathrm{id}_c)) \quad \text{from (2.3)} \\ &= T_e^0\blacktriangle(c), \end{aligned}$$

and for any arrow g in $\mathbf{C}$, and natural transformation $\triangle^1(g)$,

$$\begin{aligned} G^1\triangle^1(g) &= T_e^1(\triangle^1(g)(a_1); \triangle^1(g)(a_2)) \quad \text{from (3.27)} \\ &= T_e^1(g; g) \quad \text{from (2.4)} \\ &= T_e^1\blacktriangle^1(g) \end{aligned}$$

so that the unit of adjunction between functors $\triangle$ and G is equal to the unit η_e of the adjunction between functors $\blacktriangle$ and T_e. Moreover, for taking $\underline{f}_{_1} \triangleq \underline{f}(a_1)$ and $\underline{f}_{_2} \triangleq \underline{f}(a_2)$, we obtain from (3.27),

$$G^1(\underline{f}) = T_e^1(\underline{f}(a_1);\underline{f}(a_2)) = T_e^1(\underline{f}_{_1};\underline{f}_{_2})$$

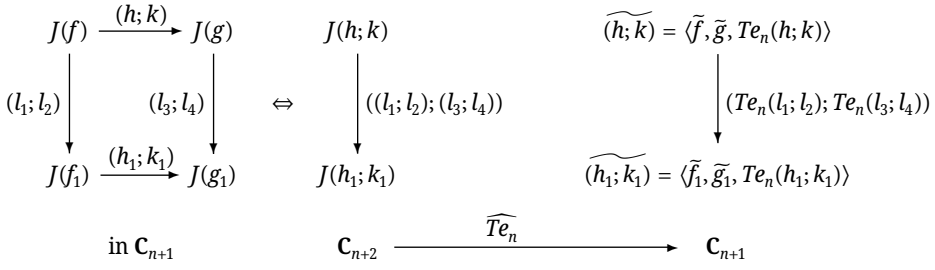
and hence the left-hand commutative diagrams of adjunction $(\blacktriangle, T_e, \varepsilon_e, \eta_e)$ in (3.23) and of new adjunction $(\triangle, G, \varepsilon, \eta_e)$ in (3.28) *are equal.*

From (3.26), and $h = d'(l_{12})$, we obtain that $\varepsilon_e(J(h)) = (\varepsilon(d')(a_1); \varepsilon(d')(a_2))$, and hence also the right-hand commutative diagrams of adjunction $(\blacktriangle, T_e, \varepsilon_e, \eta_e)$ in (3.23) and of new adjunction $(\triangle, G, \varepsilon, \eta_e)$ in (3.28) *are equivalent representations* in the categories $\mathbf{C} \downarrow \mathbf{C}$ and $\mathbf{C}^{\mathbf{J}}$, relatively. □

Finally, let us show that the CoCC property propagates to all n-dimensional levels.

Theorem 5. *Let a base category* $\mathbf{C}$ *be a conceptually closed category. Then every n-dimensional level* $\mathbf{C}_n$ *for* $n \geq 1$ *is conceptually closed, and hence this conceptual symmetry is globally valid in all n-dimensional levels.*

Proof. Let us consider the comma lifting for n-dimensional levels of the "middle projection" functor in the 1-dimensional level $Te_1 = T_e : \mathbf{C}_2 \to \mathbf{C}_1$ of the CoCC $\mathbf{C}_1$ (we recall that $\mathbf{C}_1 = \mathbf{C}$ and $\mathbf{C}_{n+1} = \mathbf{C}_n \downarrow \mathbf{C}_n$), and let us suppose that in the n-dimensional level is the functor $Te_n : \mathbf{C}_{n+1} \to \mathbf{C}_n$ and show how it propagates by the comma-lifting $\widehat{Te_n} : \mathbf{C}_{n+2} \to \mathbf{C}_{n+1}$ by the following diagram:



from the fact that the conceptualized objects of $\mathbf{C}_{n+1}$ obtained from its arrows,

$$\begin{aligned}\widetilde{(h;k)} &= \widehat{Te_n}(J(h;k)) \\ &= J(Te_n(h;k)) \\ &= \langle Te_n(J(f)), Te_n(J(g)), Te_n(h;k)\rangle \\ &= \langle \widetilde{f}, \widetilde{g}, Te_n(h;k)\rangle\end{aligned}$$

is derived from the arrows $Te_n(h;k)$ between the conceptualized objects $\widetilde{f}$ and $\widetilde{g}$ of the lower $\mathbf{C}_n$ level. Consequently, we have descending comma-induced functors $Te_{n+1} = Te_n \circ L_\perp : \mathbf{C}_{n+2} \to \mathbf{C}_{n+1}$ for all n-dimensional levels, for $n \geq 1$. In order to demonstrate that all

$\mathbf{C}_{n+1}$ levels are CoCC, it is enough to show that in all these categories we have also the isomorphic natural transformation

$$\varphi_{n+1} : Te_{n+1} \circ \blacktriangle_{n+1} \simeq \mathrm{Id}_{\mathbf{C}_{n+1}}$$

where for descending functor $Te_{n+1} = \widehat{Te_n} \circ L_\perp$ and for ascending diagonal functor $\blacktriangle_{n+1} = L_\perp \circ \widehat{\blacktriangle_n}$, and hence from the fact that $L_\perp \circ L_\perp$ is an identity, we obtain that

$$\begin{aligned} Te_{n+1} \circ \blacktriangle_{n+1} &= \widehat{Te_n} \circ L_\perp \circ L_\perp \circ \widehat{\blacktriangle_n} \\ &= \widehat{Te_n} \circ \widehat{\blacktriangle_n} \\ &= \widehat{Te_n \circ \blacktriangle_n} \end{aligned}$$

and from the fact that $\mathrm{Id}_{\mathbf{C}_{n+1}} = \widehat{\mathrm{Id}_{\mathbf{C}_n}}$, we obtain the base case of comma-induction for the natural transformation φ, i. e.,

$$\varphi_{n+1} = \widehat{\varphi_n} = (\varphi_n F^0_{\mathrm{st}}; \varphi_n S^0_{\mathrm{nd}}). \tag{3.30}$$

Thus, let by inductive hypothesis $\varphi_n : Te_n \circ \blacktriangle_n \simeq \mathrm{Id}_{\mathbf{C}_n}$ is a natural isomorphism, so that the components of natural transformations $\varphi_n \bullet \varphi_n^{-1}$ and $\varphi_n^{-1} \bullet \varphi_n$ generate the identity arrows in $\mathbf{C}_n$. Let us show that it holds also for the $(n+1)$-dimensional levels. In fact, for each object $J(f)$ of $\mathbf{C}_{n+1}$ where $f : a \to b$ is an arrow of $\mathbf{C}_n$, we have that the identity arrow of this object in $\mathbf{C}_{n+1}$ is

$$\mathrm{id}_{J(f)} = (\mathrm{id}_a; \mathrm{id}_b) : J(f) \to J(f) \tag{3.31}$$

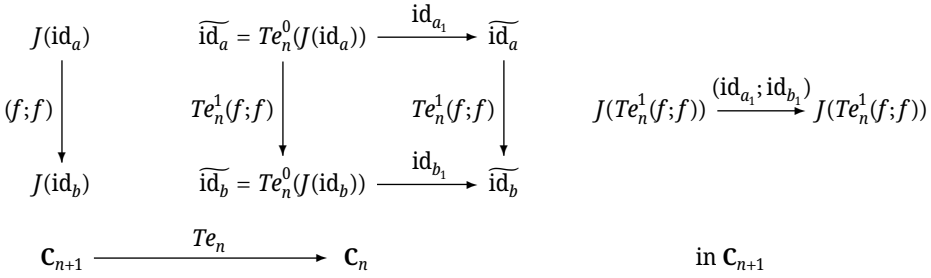
and for this object $J(f)$ we obtain the object (see Example 3 in Section 1.3 as well):

$$\begin{aligned} a_3 &\triangleq Te_{n+1} \circ \blacktriangle_{n+1}(J(f)) \\ &= \widehat{Te_n \circ \blacktriangle_n}(J(f)) \quad \text{from above} \\ &= (J(Te^1_n \circ \blacktriangle^1_n)\psi)(J(f)) \quad \text{from definition of comma lifting} \\ &= J(Te^1_n \circ \blacktriangle^1_n(f)) \quad \text{from the identity } \psi J \\ &= J(Te^1_n(f; f)). \end{aligned}$$

Let us show that the identity arrow of this object is equal to the arrow $(\mathrm{id}_{a_1}; \mathrm{id}_{b_1})$ where

$$a_1 \triangleq \widetilde{\mathrm{id}_a} = (Te_n \circ \blacktriangle_n)(a) \quad \text{and} \quad b_1 \triangleq \widetilde{\mathrm{id}_b} = (Te_n \circ \blacktriangle_n)(b) \tag{3.32}$$

by the following scheme:



and hence, from the identity arrow on the right-hand side (in $\mathbf{C}_{n+1}$), we obtain that

$$(\mathrm{id}_{a_1};\mathrm{id}_{b_1}) = \mathrm{id}_{a_3} : Te_{n+1} \circ \blacktriangle_{n+1}(J(f)) \to Te_{n+1} \circ \blacktriangle_{n+1}(J(f)). \tag{3.33}$$

Finally, we have for the $(n+1)$-dimensional level in $\mathbf{C}_{n+1}$, for any object $J(f)$, that the component of natural transformation $\varphi_{n+1}^{-1} \bullet \varphi_{n+1}$ is

$$\begin{aligned}
(\varphi_{n+1} \bullet \varphi_{n+1}^{-1})(J(f)) &= \varphi_{n+1}(J(f)) \circ \varphi_{n+1}^{-1}(J(f)) \\
&= (\varphi_n F^0_{\mathrm{st}}; \varphi_n S^0_{\mathrm{nd}})(J(f)) \circ (\varphi_n^{-1} F^0_{\mathrm{st}}; \varphi_n^{-1} S^0_{\mathrm{nd}})(J(f)) \quad \text{from (3.30)} \\
&= (\varphi_n(a); \varphi_n(b)) \circ (\varphi_n^{-1}(a); \varphi_n^{-1}(b)) \\
&= (\varphi_n(a) \circ \varphi_n^{-1}(a); \varphi_n(b) \circ \varphi_n^{-1}(b)) \\
&= ((\varphi_n \bullet \varphi_n^{-1})(a); (\varphi_n \bullet \varphi_n^{-1})(b)) \\
&= (\mathrm{id}_a; \mathrm{id}_b) \quad \text{from inductive hypothesis} \\
&= \mathrm{id}_{J(f)}
\end{aligned}$$

is equal to the identity arrow of this object $J(f)$, and the component of the natural transformation $\varphi_{n+1} \bullet \varphi_{n+1}^{-1}$ for the object $(J(f))$ is

$$\begin{aligned}
(\varphi_{n+1}^{-1} \bullet \varphi_{n+1})(J(f)) &= \varphi_{n+1}^{-1}(J(f)) \circ \varphi_{n+1}(J(f)) \\
&= (\varphi_n^{-1} F^0_{\mathrm{st}}; \varphi_n^{-1} S^0_{\mathrm{nd}})(J(f)) \circ (\varphi_n F^0_{\mathrm{st}}; \varphi_n S^0_{\mathrm{nd}})(J(f)) \quad \text{from (3.30)} \\
&= (\varphi_n^{-1}(a); \varphi_n^{-1}(b)) \circ (\varphi_n(a); \varphi_n(b)) \\
&= (\varphi_n^{-1}(a) \circ \varphi_n(a); \varphi_n^{-1}(b) \circ \varphi_n(b)) \\
&= ((\varphi_n^{-1} \bullet \varphi_n)(a); (\varphi_n^{-1} \bullet \varphi_n)(b)) \\
&= (\mathrm{id}_{a_1}; \mathrm{id}_{b_1}) \quad \text{from (3.32)} \\
&= \mathrm{id}_{a_3} \quad \text{from (3.33)}
\end{aligned}$$

is the identity arrow of the object $a_3 \triangleq Te_{n+1} \circ \blacktriangle_{n+1}(J(f))$.

Consequently, we obtain the natural isomorphism $\varphi_{n+1} : Te_{n+1} \circ \blacktriangle_{n+1} \simeq \mathrm{Id}_{\mathbf{C}_{n+1}}$ for the $(n+1)$-dimensional level as well. □

3.4 Symmetry-extended categories

By introduction of the n-dimensional levels, which are the arrow categories, we need to consider the relationship between the isomorphic arrows in the base category $\mathbf{C}$ and its arrow category $\mathbf{C}_2 = \mathbf{C} \downarrow \mathbf{C}$. Consequently, from the fact that the objects of $\mathbf{C} \downarrow \mathbf{C}$ are obtained from the arrows of $\mathbf{C}$ the isomorphic arrows in $\mathbf{C} \downarrow \mathbf{C}$ introduce the new concept of a kind of "isomorphism of arrows," together with the standard concept of an isomorphism of objects, and with this new kind of "isomorphism," we can introduce the following definition:

Definition 13. Let $h : a \to c$ and $k : b \to d$ be two arrows of $\mathbf{C}$. We tell that these two arrows h and k are "*equal up to isomorphism,*" denoted by

$$h \cong k$$

iff there is the (standard) isomorphism[3] $\mathrm{is} = (\mathrm{is}_1; \mathrm{is}_2) : J(h) \to J(k)$ (satisfying the *equality* $k \circ \mathrm{is}_1 = \mathrm{is}_2 \circ h$) of the objects $J(h)$ and $J(k)$ of the arrow category $\mathbf{C} \downarrow \mathbf{C}$.

In the previous section, we introduced the symmetry-closure functor $T_e : \mathbf{C} \downarrow \mathbf{C} \to \mathbf{C}$, which generates the conceptualized objects of $\mathbf{C}$ from its arrows, with two particular natural transformations represented by the diagram (3.17),

$$\begin{array}{ccc} & \xrightarrow{F_{\mathrm{st}}} & \\ & \downarrow \tau & \\ \mathbf{C} & \xrightarrow[S_{\mathrm{nd}}]{\downarrow \tau^{-1}} & \mathbf{C} \end{array} \quad \text{with } T_e \text{ between the two } \mathbf{C}$$

such that their vertical composition is given by (3.16),

$$\tau^{-1} \bullet \tau = (\varphi S_{\mathrm{nd}}) \bullet (T_e(\psi;\psi)) \bullet (\varphi^{-1} F_{\mathrm{st}})$$

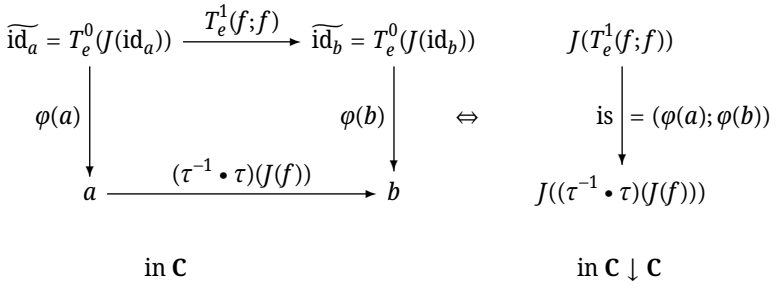
Thus, for each object $J(f)$ of $\mathbf{C} \downarrow \mathbf{C}$, obtained from the arrow $f : a \to b$ of $\mathbf{C}$, we obtain that

$$\begin{aligned} (\tau^{-1} \bullet \tau)(J(f)) &= ((\varphi S_{\mathrm{nd}}) \bullet (T_e(\psi;\psi)) \bullet (\varphi^{-1} F_{\mathrm{st}}))(J(f)) \\ &= (\varphi S_{\mathrm{nd}})(J(f)) \circ (T_e(\psi;\psi))(J(f)) \circ (\varphi^{-1} F_{\mathrm{st}})(J(f)) \\ &= \varphi(b) \circ T_e^1(f;f) \circ \varphi^{-1}(a) \end{aligned}$$

where $\varphi(a) : \widetilde{\mathrm{id}_b} \to b$ and $\varphi^{-1}(a) : a \to \widetilde{\mathrm{id}_a}$ are two isomorphic arrows in $\mathbf{C}$, (having their inverse arrows as well), so that the equation above can be represented by a commutative diagram in $\mathbf{C}$ corresponding to an isomorphic arrow in $\mathbf{C} \downarrow \mathbf{C}$ as follows:

---

3 It is easy to verify that each isomorphic arrow in $\mathbf{C} \downarrow \mathbf{C}$ is a pair of isomorphic arrows of $\mathbf{C}$. Thus, in this case we have two isomorphic arrows in $\mathbf{C}$: $\mathrm{is}_1 : a \to b$ and $\mathrm{is}_2 : c \to d$.



where is : $J(T_e^1(f;f)) \to J((\tau^{-1} \bullet \tau)(J(f)))$ is an isomorphic arrow in $\mathbf{C} \downarrow \mathbf{C}$, so that, from Definition 13, we have the following two arrows "equal up to isomorphism" in $\mathbf{C}$:

$$T_e^1(f;f) \cong (\tau^{-1} \bullet \tau)(J(f)). \tag{3.34}$$

So, the specific interesting case of this general result is when $(\tau^{-1} \bullet \tau)(J(f)) = f$, so that $T_e^1(f;f)$ is equal up to isomorphism to the arrow f.

Remark. Note that this isomorphism

$$T_e^1(f;f) \cong f$$

can be obtained directly from (3.12), i. e., $T_e^1(f;f) = \varphi^{-1}(\mathrm{cod}(f)) \circ f \circ \varphi(\mathrm{dom}(f))$, so that $\varphi(\mathrm{cod}(f)) \circ T_e^1(f;f) = \varphi(\mathrm{cod}(f)) \circ \varphi^{-1}(\mathrm{cod}(f)) \circ f \circ \varphi(\mathrm{dom}(f)) = f \circ \varphi(\mathrm{dom}(f))$, that is to the arrow $(\varphi(\mathrm{cod}(f); \varphi(\mathrm{dom}(f)) : J(T_e^1(f;f)) \to J(f)$ in $\mathbf{C} \downarrow \mathbf{C}$ where both $\varphi(\mathrm{cod}(f))$ and $\varphi(\mathrm{dom}(f))$ are the isomorphisms in $\mathbf{C}$. □

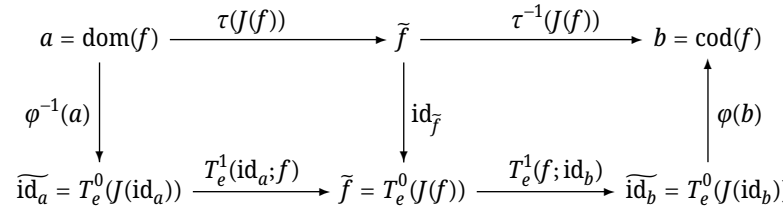
In this case, we have the following kind of categorial symmetry.

Definition 14. A *symmetry-extended* category $\mathbf{C}$ is CoCC, such that each its arrow f is equal up to isomorphism to the arrow $T_e^1(f;f)$, i. e., $T_e^1(f;f) \cong f$, and hence with

$$\tau^{-1} \bullet \tau = \psi : F_{\mathrm{st}} \xrightarrow{\cdot} S_{\mathrm{nd}} \tag{3.35}$$

for vertical composition of natural transformations $\tau : F_{\mathrm{st}} \xrightarrow{\cdot} T_e$ and $\tau^{-1} : T_e \xrightarrow{\cdot} S_{\mathrm{nd}}$ defined by (3.15).

Note that from the commutative diagram (3.19) in $\mathbf{C}$,

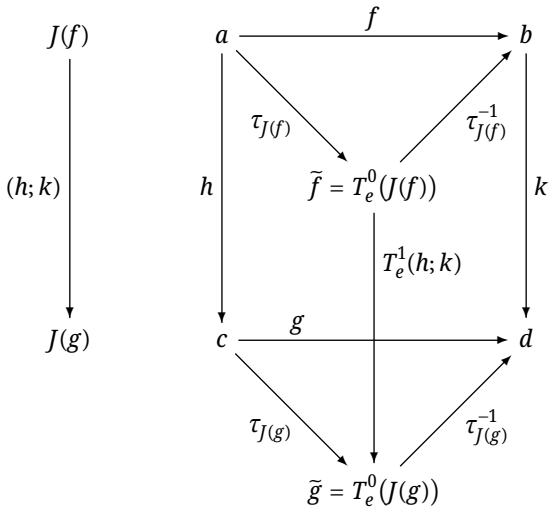


all vertical arrows are isomorphic arrows in $\mathbf{C}$, so that left and right commutative subdiagrams are represented by two isomorphic arrows of $\mathbf{C} \downarrow \mathbf{C}$, $(\varphi^{-1}(a); \mathrm{id}_{\tilde{f}}) : J(\tau(J(f))) \to$

$J(T^1_e(\mathrm{id}_a; f))$ and $(\mathrm{id}_{\widetilde{f}}; \varphi(b)) : J(T^1_e(f; \mathrm{id}_b)) \to J(\tau^{-1}(J(f)))$, and hence we obtained, in this symmetry-extended category $\mathbf{C}$, the following "equal up to isomorphism" arrows, for each given arrow f,

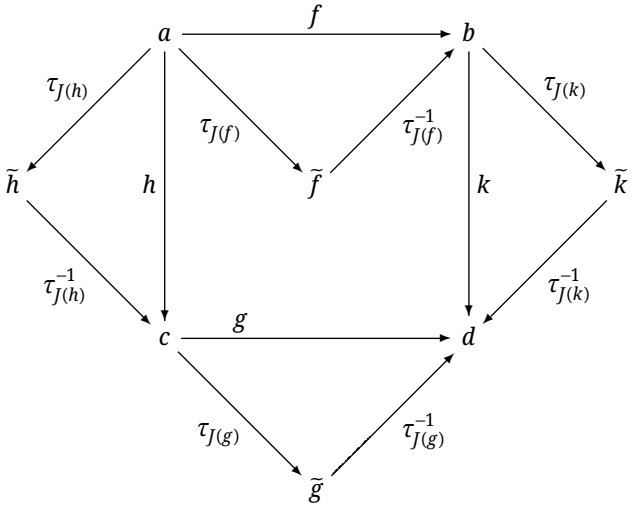
$$\tau(J(f)) \cong T^1_e(\mathrm{id}_{\mathrm{dom}(f)}; f) \quad \text{and} \quad \tau^{-1}(J(f)) \cong T^1_e(f; \mathrm{id}_{\mathrm{cod}(f)}). \tag{3.36}$$

So, in a symmetry-extended category $\mathbf{C}$, for each arrow $(h; k) : J(f) \to J(g)$ (an equation $k \circ f = g \circ h$) in $\mathbf{C} \downarrow \mathbf{C}$, we obtain the following commutative diagram in $\mathbf{C}$:



(3.37)

which represents the *strong correlation* between an arrow and its conceptualized object: each arrow is equal to the composition of two arrows with their "middle" object equal to the conceptualized object of this original arrow, i. e.,



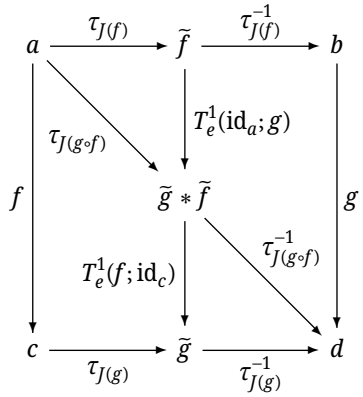
and this decomposition can inductively continue for the arrow components of two natural transformations τ and τ^{-1} as well.

In the diagram above, if the objects a, b, c and d belong to an n-dimensional level then the objects $J(f), J(g), J(h), J(k)$ and $J(k \circ f)$ belong to next higher $(n+1)$-dimensional level, and hence $\widetilde{f}, \widetilde{g}, \widetilde{h}, \widetilde{k}$ and $\widetilde{k \circ f} = \widetilde{k} * \widetilde{f}$ are derived objects from this higher level. Consequently, the components of the natural transformation τ we nominate as "ascending" arrows, while the components of the natural transformation τ^{-1} we nominate as "descending" arrows of the n-dimensional level. So, in each symmetry-extended category we can separate in a number of subcategories composed by the objects of the same level of conceptualization, beginning from the first level of basic (nonconceptualized) objects of such a category.

Let us now consider the conceptualized objects obtained from a composed arrow $g \circ f$ with a conceptualized composed object

$$\begin{aligned}
\widetilde{g \circ f} &= T_e^0(J(g \circ f)) \\
&= B_{\top}\psi(J(g \circ f)) \\
&= B_{\top}(g \circ f) \quad \text{from identity } \psi J \\
&= B_{\top}(g) * B_{\top}(f) \quad \text{from (3.5)} \\
&= T_e^0(J(g)) * T_e^0(J(f)) \\
&= \widetilde{g} * \widetilde{f}
\end{aligned}$$

and for that consider the diagram (3.37) when $k = g$ and $h = f$, so from it we obtain the following commutative diagram:



where the external square diagram is a trivial commutative diagram $g \circ f = g \circ f$; the internal vertical arrow is $T_e^1(f; g) = T_e^1(f; \mathrm{id}_c) \circ T_e^1(\mathrm{id}_a; g) : \widetilde{f} \to \widetilde{g}$ as in diagram (3.37),

while for the internal upper triangle diagram we have that

$$\begin{aligned}
T^1_e(\mathrm{id}_a; g) \circ \tau_{J(f)} &= T^1_e(\mathrm{id}_a; g) \circ \tau(J(f)) \\
&= T^1_e(\mathrm{id}_a; g) \circ ((T^1_e\sigma) \bullet (\varphi^{-1}F^0_{\mathrm{st}}))(J(f)) \\
&= T^1_e(\mathrm{id}_a; g) \circ ((T^1_e\sigma)(J(f)) \circ (\varphi^{-1}F^0_{\mathrm{st}})(J(f))) \\
&= T^1_e(\mathrm{id}_a; g) \circ ((T^1_e(\psi\blacktriangle^0 F^0_{\mathrm{st}}; \psi))(J(f)) \circ \varphi^{-1}(a)) \\
&= T^1_e(\mathrm{id}_a; g) \circ ((T^1_e(\psi\blacktriangle^0(a); f)) \circ \varphi^{-1}(a)) \\
&= T^1_e(\mathrm{id}_a; g) \circ ((T^1_e(\mathrm{id}_a; f)) \circ \varphi^{-1}(a)) \\
&= T^1_e(\mathrm{id}_a; g \circ f) \circ \varphi^{-1}(a) \\
&= T^1_e(\mathrm{id}_a; g \circ f) \circ (\varphi^{-1}F^0_{\mathrm{st}})(J(g \circ f)) \\
&= (T^1_e\sigma)(J(g \circ f)) \circ (\varphi^{-1}F^0_{\mathrm{st}})(J(g \circ f)) \\
&= ((T^1_e\sigma) \bullet (\varphi^{-1}F^0_{\mathrm{st}}))(J(g \circ f)) \\
&= \tau(J(g \circ f)) = \tau_{J(g\circ f)}.
\end{aligned}$$

Example 10. The **Set** is a symmetry-extended category: given any function $f : a \longrightarrow b$, the conceptualized object of this function is the graph of this function (which is a set),

$$\widetilde{f} = B_{\top}(f) \triangleq \{(x, f(x)) \mid x \in a\}.$$

The composition of objects $*$ is defined as an associative composition of binary relations (graphs of functions), $B_{\top}(g \circ f) = \{(x, (g \circ f)(x)) \mid x \in a\} = \{(y, g(y)) \mid y \in b\} \circ \{(x, f(x)) \mid x \in a\} = B_{\top}(g) * B_{\top}(f)$.

Set is conceptually closed by the functor T_e, such that for any object $J(f) = \langle a, b, f\rangle$, $\widetilde{f} = T^0_e(J(f)) = B_{\top}(f) = \{(x, f(x)) \mid x \in a\}$, and for any arrow $(k_1; k_2) : J(f) \to J(g)$, the component T^1_e is defined for any $(x, f(x)) \in T^0_e(J(f))$ by

$$T^1_e(k_1; k_2)(x, f(x)) = (k_1(x), k_2(f(x))).$$

It is easy to verify the compositional property for T^1_e, and that $T^1_e(\mathrm{id}_a; \mathrm{id}_b) = \mathrm{id}_{T^0_e(J(f))}$. The **Set** is also an extended symmetric category, such that for any object $J(f) = \langle a, b, f\rangle$ in **Set** $\downarrow$ **Set**,

– $\tau(J(f)) : a \twoheadrightarrow \widetilde{f}$ is an epimorphism, such that for any $x \in a$, $\tau(J(f))(x) = (x, f(x))$.
– $\tau^{-1}(J(f)) : \widetilde{f} \hookrightarrow b$ is the second projection. Thus, for any $(x, f(x)) \in \widetilde{f}$,

$$\tau^{-1}(J(f))(x, f(x)) = f(x).$$

Thus, each arrow in **Set** is a composition of an epimorphism (surjective function) and a monomorphism (injective function).

Another example of a symmetry-extended category, considered in detail in the next chapter, is the Natural Deduction Category **ND** for constructive interpretation of logic implication, introduced in Example 7 at the end of Section 3.1. In this **ND** category, between two objects a and b (two true formulae) there is *at maximum one* arrow representing equivalence class of all deductions of one object from another one. So, for any valid deduction $f : a \to b$, it must hold that $(\tau^{-1} \bullet \tau)(J(f)) = f$.

A significant example of such categorial symmetry, applied to database mapping, is the **DB** category, developed in [5] and introduced here in Section A.3.

Now we can show that for the n-dimensional transformations this symmetry-extended property is an invariant property valid globally in all n-dimensional levels.

Proposition 9. *Let a base category* **C** *be a symmetry-extended category. Then every n-dimensional level* $\mathbf{C}_n$*, for* $n \geq 2$ *is a symmetry-extended category, and hence this conceptual symmetry is globally valid.*

Proof. If **C** is a symmetry-extended category, then it is CoCC category. So, from the fact that this CoCC property remains invariant under n-dimensional transformations, as shown by Theorem 5 in the previous section, it is enough to show that equation (3.35), $\tau^{-1} \bullet \tau = \psi : F_{\text{st}} \xrightarrow{\cdot} S_{\text{nd}}$, is valid for all n-dimensional levels. Let us suppose that $\tau_k^{-1} \bullet \tau_k = \psi_k$ for some $k \geq 1$ and show that also for $k+1$ it holds that $\tau_{k+1}^{-1} \bullet \tau_{k+1} = \psi_{k+1}$. Note that τ, τ^{-1} and ψ are all descending cases, and hence for each of them we have the same comma-induction forms, e. g., $\tau_{k+1} = \widehat{\tau_k} L_\perp$. Consequently, we have for any object $J(f)$ of $\mathbf{C}_{n+2}$ with

$$J(g) = \langle a, b, g \rangle \triangleq L_\perp(J(f)) \tag{3.38}$$

so that

$$\begin{aligned}
(\tau_{k+1}^{-1} \bullet \tau_{k+1})(J(f)) &= (\widehat{\tau_k^{-1}} L_\perp \bullet \widehat{\tau_k} L_\perp)(J(f)) \\
&= (\widehat{\tau_k^{-1}} \bullet \widehat{\tau_k})(L_\perp J(f)) \\
&= (\widehat{\tau_k^{-1}} \bullet \widehat{\tau_k})(J(g)) \quad \text{from (3.38)} \\
&= \widehat{\tau_k^{-1}}(J(g)) \circ \widehat{\tau_k}(J(g)) \\
&= (\tau_k^{-1} F_{\text{st}}^0; \tau_k^{-1} S_{\text{nd}}^0)(J(g)) \circ (\tau_k F_{\text{st}}^0; \tau_k S_{\text{nd}}^0)(J(g)) \\
&= (\tau_k^{-1}(a); \tau_k^{-1}(b)) \circ (\tau_k(a); \tau_k(b)) \\
&= (\tau_k^{-1}(a) \circ \tau_k(a); \tau_k^{-1}(b) \circ \tau_k(b)) \\
&= ((\tau_k^{-1} \bullet \tau_k)(a); (\tau_k^{-1} \bullet \tau_k)(b)) \\
&= (\psi_k(a); \psi_k(b)) \quad \text{from inductive hypothesis} \\
&= (\psi_k F_{\text{st}}^0; S_{\text{nd}}^0)(J(g))
\end{aligned}$$

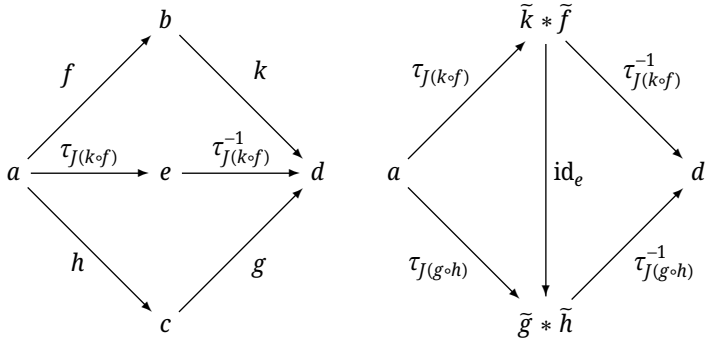
$$
\begin{aligned}
&= \widehat{\psi_k}(J(g)) \\
&= \widehat{\psi_k} L_\perp(J(f)) \quad \text{from (3.38)} \\
&= \psi_{k+1}(J(f))
\end{aligned}
$$

and hence, from the fact that it holds for any object $J(f)$ in $\mathbf{C}_{n+2}$, we obtain

$$\tau_{k+1}^{-1} \bullet \tau_{k+1} = \psi_{k+1}. \qquad \square$$

Let us show the statement of Definition 11 (in footnotes): While the equation in category $\mathbf{C}$ can be represented by commutative diagrams composed by the arrows of the category, the corresponding equations represented by a composition of the objects cannot be represented in the same category because both objects on left and right side of an equation would be represented by the same unique object in category with its identity arrow. So, the equations of composition of the objects has to be supported in a different category $\widetilde{\mathbf{C}}$ in (3.11), or standard for the mathematical system of equations.

For example, for a given equation of arrow compositions, $k \circ f = g \circ h$, we have in an extended-symmetry category $\mathbf{C}$ the following two representations, left-hand side by commutative diagrams of these arrows, while on the right-hand side this equation e_O between composition of objects is given by identity arrow $\mathrm{id}_{e_O} \triangleq T_e^1(\mathrm{id}_a; \mathrm{id}_d)$, while both square commutative diagrams represent the same equation $k \circ f = g \circ h$, from the fact that $\tau_{J(k\circ f)}^{-1} \circ \tau_{J(k\circ f)} = k \circ f$ and $\tau_{J(g\circ h)}^{-1} \circ \tau_{J(g\circ h)} = g \circ h$,



However, this identity arrow id_e representing the equation e_O,

$$\widetilde{k} * \widetilde{f} = \widetilde{g} * \widetilde{h} \tag{3.39}$$

as an analog to the commutative diagrams in a given category, corresponding to the unique object in a this category, so that both composed objects $\widetilde{k} * \widetilde{f}$ and $\widetilde{g} * \widetilde{h}$ are represented by the *same* object, say the first one $e \triangleq \widetilde{k} * \widetilde{f}$ (in the left-hand side diagram), and

that is reason that the equation (3.39) cannot be represented by the same category **C** but only in $\widetilde{\mathbf{C}}$ as presented in (3.11). In fact, the whole right-hand side diagram above in this category is reduced into the composed horizontal arrow in the left-hand side diagram above.

That is, the equations represented by composition of objects as an analog representation to the commutative diagrams in a given category **C**, needs another mathematical category $\widetilde{\mathbf{C}}$: this is the proper meaning of the *technically equivalent representation* of the equations composed by arrows and that are composed by their conceptualized objects.

3.5 Symmetry hierarchy upper bound: Imploded categories

This chapter, dedicated to the internal categorial symmetry of its primitive categorial concepts, arrows and objects, and to the classification (hierarchy) of different kinds of these symmetries, we will conclude with the extreme limit case of the symmetry-extended categories analyzed in the previous section. However, this limit case is important only from a theoretical point of view, because we cannot impose more constraints than that used for this limit case, in order to obtain another subclass of symmetric categories.

From the more concrete point of view, this limit case of symmetric categories is practically insignificant, as we will show, because such categories have the very poor structural properties.

Just because of that, before a specification of this bottom level in the category symmetry hierarchy, as an important conclusion about internal categorial symmetry and its classification, it is useful to consider the symmetry-extended categories as the most expressive level of category symmetry, with most constraints that preserves its symmetry property, but still with great capability to represent a lot of fundamental mathematical structures.

In fact, we have shown in the previous section, in Example 10, that also the **Set** category is a symmetry-extended category, and this category is fundamental for the denotational semantics in a significant number of mathematical theories.

So, let us consider how the conceptualized objects (obtained from the arrows), in a given base symmetry-extended category **C**, are derived not only from its own arrows, but also from the arrows of higher n-dimensional levels $\mathbf{C}_n$, for $n \geq 2$.

Example 11. Let us consider the minimal case (for $n = 2$) of composed functors $Te_2 \circ T_e$: $\mathbf{C}_3 \rightarrow \mathbf{C}$ and let us consider the arrow $((l_1; l_2); (l_3; l_4)) : J(h_1; k_1) \rightarrow J(h_2; k_2)$ of $\mathbf{C}_3$, with two the arrows $(h_1; k_1) : J(f_1) \rightarrow J(g_1)$ and $(h_2; k_2) : J(f_2) \rightarrow J(g_2)$ in $\mathbf{C}_2 = \mathbf{C} \downarrow \mathbf{C}$, so that by

application of the descending functor $Te_2 = \widehat{T_e} \circ L_\perp$,

$$Te_2 = \widehat{T_e} \circ L_\perp = (JT_e^1\psi; (T_e^1 F_{\mathrm{st}}^1; T_e^1 S_{\mathrm{nd}}^1)) \circ L_\perp \tag{3.40}$$

to this arrow in $\mathbf{C}_3$, we obtain for the imploded category $\mathbf{C}_2$, with descending natural transformations

$$\tau_2 = \widehat{\tau} L_\perp = (\tau F_{\mathrm{st}}^0; \tau S_{\mathrm{nd}}^0) L_\perp \quad \text{and} \quad \tau_2^{-1} = \widehat{\tau^{-1}} L_\perp = (\tau^{-1} F_{\mathrm{st}}^0; \tau^{-1} S_{\mathrm{nd}}^0) L_\perp \tag{3.41}$$

with the functor $L_\perp = (L_\perp^0, L_\perp^1) : \mathbf{C}_3 \to \mathbf{C}_3$, with object's and arrow's components derived from (1.10),

$$L_\perp^0 = J\widehat{\psi} = J(\psi F_{\mathrm{st}}^0; \psi S_{\mathrm{nd}}^0) \quad \text{and} \quad L_\perp^1 = ((F_{\mathrm{st}}^1 F_{\mathrm{st}}^1; F_{\mathrm{st}}^1 S_{\mathrm{nd}}^1); (S_{\mathrm{nd}}^1 F_{\mathrm{st}}^1; S_{\mathrm{nd}}^1 S_{\mathrm{nd}}^1)). \tag{3.42}$$

Thus, we have that

$$\begin{aligned} Te_2^0(J(h_1; k_1)) &= \widehat{T_e} L_\perp^0 (J(h_1; k_1)) \\ &= \widehat{T_e} J(\psi F_{\mathrm{st}}^0; \psi S_{\mathrm{nd}}^0)(J(h_1; k_1)) \\ &= \widehat{T_e} J(f_1; g_1) \\ &= JT_e^1 \psi J(f_1; g_1) \end{aligned}$$

i. e.,

$$Te_2^0(J(h_1; k_1)) = J(T_e^1(f_1; g_1)) \tag{3.43}$$

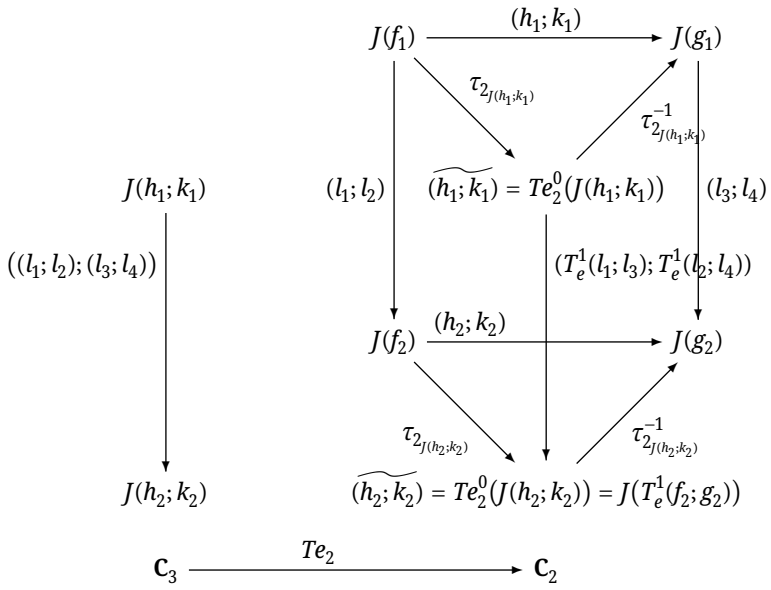
and

$$\begin{aligned} Te_2^1((l_1; l_2); (l_3; l_4)) &= \widehat{T_e} L_\perp^1 ((l_1; l_2); (l_3; l_4)) \\ &= \widehat{T_e}((l_1; l_3); (l_2; l_4)) \\ &= (T_e^1 F_{\mathrm{st}}^1; T_e^1 S_{\mathrm{nd}}^1)((l_1; l_3); (l_2; l_4)) \end{aligned}$$

i. e.,

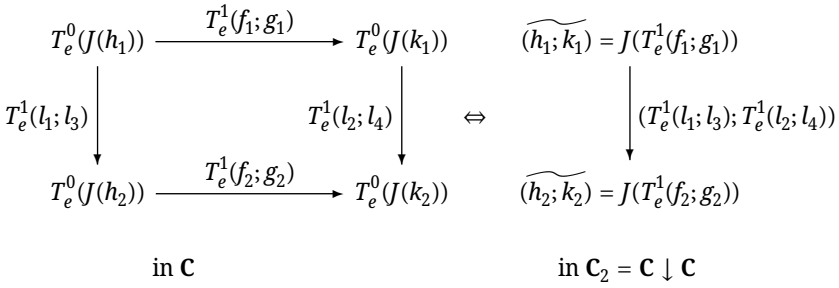
$$Te_2^1((l_1; l_2); (l_3; l_4)) = (T_e^1(l_1; l_3); T_e^1(l_2; l_4)) \tag{3.44}$$

and we obtain the following commutative diagram in $\mathbf{C}_2$:



(3.45)

where the square commutative diagram in $\mathbf{C}_2$ corresponds to the vertical arrow in $\mathbf{C}_3$ (which is a commutative cube in $\mathbf{C}$), and the middle vertical arrow $(T^1_e(l_1;l_3);T^1_e(l_2;l_4))$ in $\mathbf{C}_2$, obtained by application of the comma-inducted functor $Te_2 : \mathbf{C}_3 \to \mathbf{C}_2$ on the vertical arrow $((l_1;l_2);(l_3;l_4))$ in $\mathbf{C}_3$, corresponds to the following commutative diagram in base category $\mathbf{C}$:



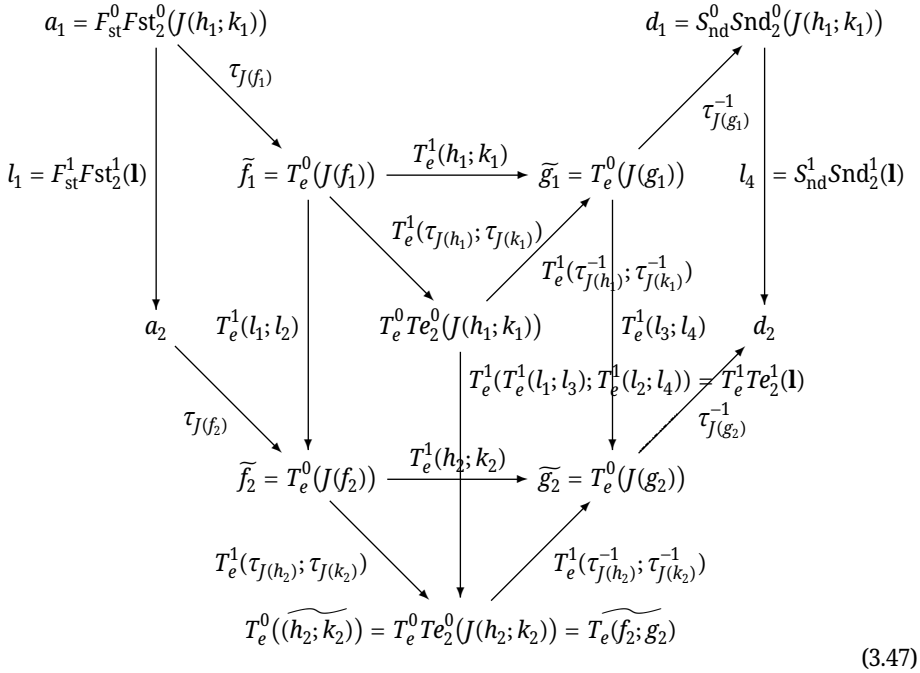
Now, in the second step, we have to apply the functor $T_e : \mathbf{C}_2 \to \mathbf{C}$ to the commutative diagram (3.45), and by considering that the arrow components obtained from the comma-induced natural transformations τ_2 and τ_2^{-1} given by (3.41),

$$\begin{aligned}
\tau_2(J(h_1;k_1)) &= (\tau F^0_{\mathrm{st}};\tau S^0_{\mathrm{nd}})L^0_{\perp}(J(h_1,k_1)) \\
&= (\tau F^0_{\mathrm{st}};\tau S^0_{\mathrm{nd}})(J\hat{\psi})(J(h_1,k_1)) \\
&= (\tau F^0_{\mathrm{st}};\tau S^0_{\mathrm{nd}})(J(f_1;g_1)) \\
&= (\tau(J(h_1));\tau(J(k_1)))
\end{aligned}$$

i. e.,

$$\tau_2(J(h_1;k_1)) = (\tau_{J(h_1)};\tau_{J(k_1)}) \tag{3.46}$$

so that we obtain the following commutative diagram in $\mathbf{C}$, by defining also the arrow $\mathbf{l} \triangleq ((l_1;l_2);(l_3;l_4)) : J(h_1;k_1) \to J(h_2;k_2)$ in $\mathbf{C}_3$:



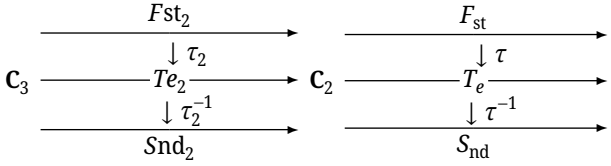
(3.47)

from which we can see the conceptualized objects $\widetilde{f_i}$ and $\widetilde{g_i}$, for $i = 1,2$, obtained from the 2-dimensional level $\mathbf{C}_2$ and the conceptualized objects,

$$\widetilde{T_e(f_1;g_1)} = T^0_eTe^0_2(J(h_1;k_1)) \quad \text{and} \quad \widetilde{T_e(f_2;g_2)} = T^0_eTe^0_2(J(h_2;k_2)),$$

obtained from the higher 3-dimensional level $\mathbf{C}_3$.

Let us now consider the composition of the natural transformations represented by this diagram,



(3.48)

with two horizontal compositions,

$$(\tau \circ \tau_2) : F_{\text{st}}\text{Fst}_2 \xrightarrow{\cdot} T_e Te_2 \quad \text{and} \quad (\tau^{-1} \circ \tau_2^{-1}) : T_e Te_2 \xrightarrow{\cdot} S_{\text{nd}}\text{Snd}_2, \tag{3.49}$$

and hence for the object $J(h_1; k_1)$ of $\mathbf{C}_3$, we obtain

$$\begin{aligned}
&F\text{st}_2^0(J(h_1;k_1)) = \widehat{F_{\text{st}}}L_\perp^0(J(h_1;k_1)) \quad \text{for comma-induced descending functors}\\
&\widehat{F_{\text{st}}}(J(\psi F_{\text{st}}^0; \psi S_{\text{nd}}^0))(J(h_1;k_1)) \quad \text{from (3.42)}\\
&\quad = \widehat{F_{\text{st}}}J(f_1;g_1)\\
&\quad = JF_{\text{st}}^1\psi(J(f_1;g_1)) \quad \text{from comma-lifting}\\
&\quad = J(F_{\text{st}}^1(f_1;g_1)), \quad \text{from identity } \psi J
\end{aligned}$$

and hence

$$F\text{st}_2^0(J(h_1;k_1)) = J(f_1) \tag{3.50}$$

and analogously,

$$S\text{nd}_2^0(J(h_1;k_1)) = J(g_1) \tag{3.51}$$

so that

$$\begin{aligned}
(\tau \circ \tau_2)(J(h_1;k_1)) &= T_e^1\tau_2(J(h_1;k_1)) \circ \tau(F\text{st}_2^0(J(h_1;k_1))) \quad \text{from (A.12)}\\
&= T_e^1(\tau_{J(h_1)}; \tau_{J(k_1)}) \circ \tau(F\text{st}_2^0(J(h_1;k_1))) \quad \text{from (3.46)}\\
&= T_e^1(\tau_{J(h_1)}; \tau_{J(k_1)}) \circ \tau(J(f_1)) \quad \text{from (3.50)}\\
&= T_e^1(\tau_{J(h_1)}; \tau_{J(k_1)}) \circ \tau_{J(f_1)} : F_{\text{st}}^0 F\text{st}_2^0(J(h_1;k_1)) \to T_e^0 Te_2^0(J(h_1;k_1)),
\end{aligned}$$

which is the left-upper composed arrow in diagram (3.47). Analogously,

$$\begin{aligned}
(\tau^{-1} \circ \tau_2^{-1})(J(h_1;k_1)) &= \tau^{-1}(S\text{nd}_2^0(J(h_1;k_1))) \circ T_e^1\tau_2^{-1}(J(h_1;k_1)) \quad \text{from (A.12)}\\
&= \tau^{-1}(S\text{nd}_2^0(J(h_1;k_1))) \circ T_e^1(\tau_{J(h_1)}^{-1}; \tau_{J(k_1)}^{-1}) \quad \text{from (3.46)}\\
&= \tau^{-1}(J(g_1)) \circ T_e^1(\tau_{J(h_1)}^{-1}; \tau_{J(k_1)}^{-1}) \quad \text{from (3.51)}\\
&= \tau_{J(g_1)}^{-1} \circ T_e^1(\tau_{J(h_1)}^{-1}; \tau_{J(k_1)}^{-1}) : T_e^0 Te_2^0(J(h_1;k_1)) \to S_{\text{nd}}^0 S\text{nd}_2^0(J(h_1;k_1)),
\end{aligned}$$

which is the right-upper composed arrow in diagram (3.47), and hence, the whole upper diagrams arrow from object a_1 to d_1 is equal to composition $(\tau^{-1} \circ \tau_2^{-1})(J(h_1;k_1)) \circ (\tau \circ \tau_2)(J(h_1;k_1))$, i. e., to the vertical composition of two horizontal compositions, i. e., we obtain the upper composed arrow of the diagram (3.47) equal to

$$((\tau^{-1} \circ \tau_2^{-1}) \bullet (\tau \circ \tau_2))(J(h_1;k_1)) : a_1 \to d_1 \tag{3.52}$$

with (composed) natural transformation, in accordance with scheme (3.48),

$$((\tau^{-1}\circ\tau_2^{-1})\bullet(\tau\circ\tau_2)) : F_{\mathrm{st}}F\mathrm{st}_2 \xrightarrow{\cdot} S_{\mathrm{nd}}S\mathrm{nd}_2$$

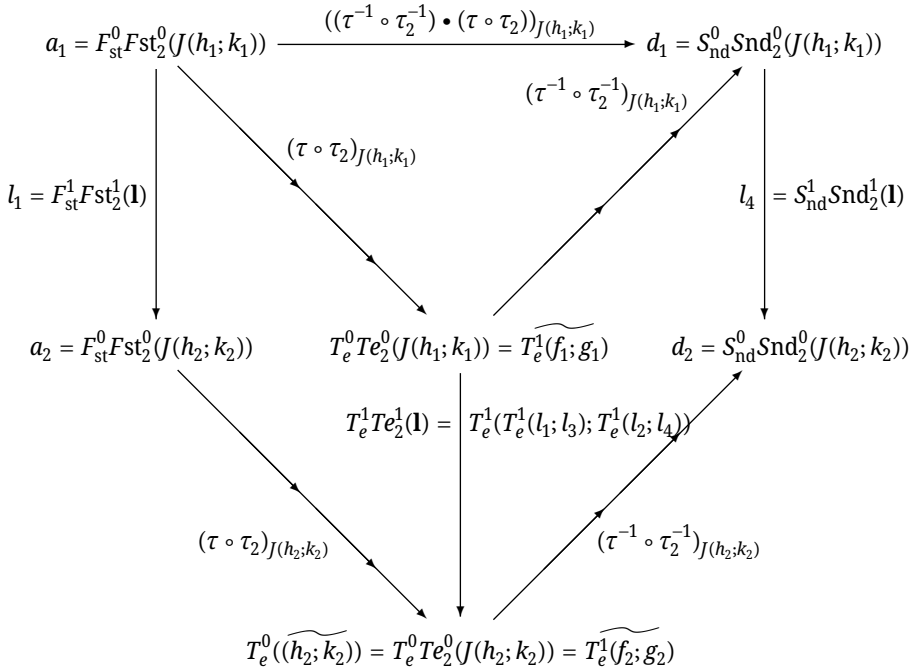
so that the bottom composed arrow of the diagram (3.47) is equal to

$$((\tau^{-1}\circ\tau_2^{-1})\bullet(\tau\circ\tau_2))(J(h_2;k_2)) : a_2 \to d_2. \tag{3.53}$$

Moreover,

$$\begin{aligned}
F^1_{\mathrm{st}}F\mathrm{st}^1_2(\mathbf{l}) &= F^1_{\mathrm{st}}F\mathrm{st}^1_2((l_1;l_2);(l_3;l_4))\\
&= F^1_{\mathrm{st}}(\widehat{F_{\mathrm{st}}L^1_\perp})((l_1;l_2);(l_3;l_4)) \quad \text{from comma-induction of descending functor}\\
&= F^1_{\mathrm{st}}\widehat{F_{\mathrm{st}}}((l_1;l_3);(l_2;l_4)) \quad \text{from (3.42)}\\
&= F^1_{\mathrm{st}}(F^1_{\mathrm{st}}F^1_{\mathrm{st}};F^1_{\mathrm{st}}S^1_{\mathrm{nd}})((l_1;l_3);(l_2;l_4)) \quad \text{from comma-lifting of a functor}\\
&= F^1_{\mathrm{st}}(l_1;l_2)((l_1;l_3);(l_2;l_4))\\
&= l_1
\end{aligned}$$

as shown in diagram (3.47). In the same way, is easy to verify that $S^1_{\mathrm{nd}}S\mathrm{nd}^1_2(\mathbf{l}) = l_4$. So, this diagram (3.47) really commutes, and can be shortly presented by this commutative diagram:



So, we can continue to consider also how we obtain the conceptualized objects in the base symmetry-extended category $\mathbf{C}$, obtained from the arrows in the $\mathbf{C}_3$ level of its categories, obtained by the composed functor $T_e \circ Te_2 \circ Te_3 : \mathbf{C}_4 \to \mathbf{C}$, like the conceptualized object $T_e^0 Te_2^0 Te_3^0(J(\mathbf{1})) = T_e^0 Te_2^0 Te_3^0(J((l_1; l_2); (l_3; l_4)))$, etc.

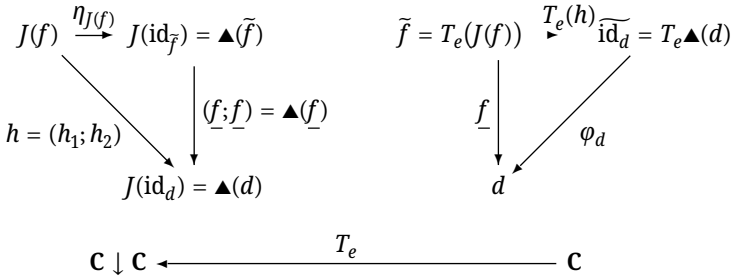
Now we can define the limit case (with the most constraints) of the internal categorial symmetry, and successively, to provide the final hierarchy of the classes of categories with internal symmetry, and their relationship with n-dimensional levels as well.

As we will show, the extreme case of internal categorial symmetry corresponds to the equivalence between all n-dimensional levels and basic category $\mathbf{C}$. In this extreme case, basic category is denominated as "imploded category" from the fact that all n-dimensional levels are flattened (and contained) inside this basic category.

Formal specification of this extreme case of internal categorial symmetry is provided by the following definition.

Definition 15. An imploded category $\mathbf{C}$ is a CoCC (Definition 14) for which there exists the adjoint equivalence $(T_e, \blacktriangle, \varphi, \eta)$ of the closure functor $T_e : \mathbf{C} \downarrow \mathbf{C} \to \mathbf{C}$, and its right adjoint diagonal functor $\blacktriangle : \mathbf{C} \to \mathbf{C} \downarrow \mathbf{C}$.

So, in this definition of adjunction, the counit is a natural isomorphism φ defined in (3.13), while the unit is new natural isomorphism $\eta : \mathrm{Id}_{\mathbf{C}\downarrow\mathbf{C}} \to \blacktriangle \circ T_e$, with the following adjunction diagram for each $J(f)$ of a given arrow $f : a \to b$ in $\mathbf{C}$ and object $\mathbf{d}$ of C,



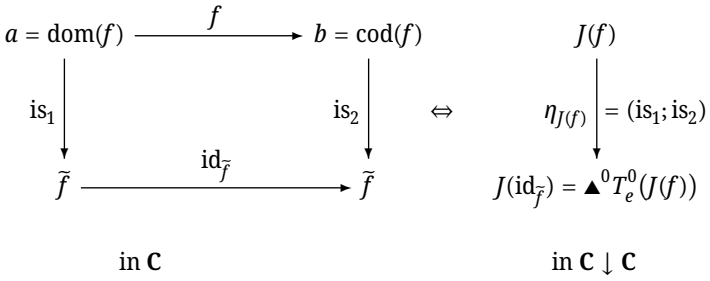
(3.54)

Thus, for any object $J(f)$ of $\mathbf{C} \downarrow \mathbf{C}$, the pair $(\tilde{f}, \eta_{J(f)})$ is an universal arrow, so that for any other pair (d, h) there exist a unique arrow $\underline{f}$ from $\tilde{f}$ to d in $\mathbf{C}$.

Note that the isomorphic arrow (an arrow component of the natural isomorphism η), $(\mathrm{is}_1; \mathrm{is}_2) = \eta_{J(f)} : J(f) \to J(\mathrm{id}_{\tilde{f}})$ of $\mathbf{C} \downarrow \mathbf{C}$ is composed by two isomorphic arrows,

$$\mathrm{is}_1 : a \simeq \tilde{f} \quad \text{and} \quad \mathrm{is}_2 : b \simeq \tilde{f}, \tag{3.55}$$

that compose the following commutative diagram:



(3.56)

Hence, from the commutative diagram with two isomorphic arrows is_1 and is_2, we obtain that

$$f = \mathrm{is}_2^{-1} \circ \mathrm{is}_1 \tag{3.57}$$

i. e., the arrow f must be an isomorphism, $f : a \simeq b$, as well. That is, the commutative diagram in (3.56) represents the isomorphisms

$$\mathrm{dom}(f) \simeq \tilde{f} \simeq \mathrm{cod}(f) \tag{3.58}$$

for every arrow of **C**. From the fact that it holds for every object $J(f)$ of $\mathbf{C} \downarrow \mathbf{C}$, we have that the arrows of **C**, different from the identity arrows, *must be isomorphic arrows*.

Based on this result that the arrows of an imploded category are identity arrows or isomorphic arrows, we have the following result.

Theorem 6. *Each imploded category is a symmetry-extended category and all its n-dimensional levels, for $n \geq 2$, are imploded categories as well, and hence this conceptual symmetry is globally valid.*

Proof. If a given base category **C** is an imploded category with adjunction $(T_e, \blacktriangle, \varphi, \eta)$ in Definition 15, then the comma-propagation of this adjunction, from Theorem 2 in Section 1.4, is valid in each n-dimensional level $\mathbf{C}_n$, for $n \geq 1$, and hence each of them is an imploded category as well.

Let us show that an imploded category **C** is symmetry-extended category as well. From the fact that an imploded category is by definition a conceptually closed category and that its counit is, from (3.13), the natural isomorphism

$$\varphi : T_e \circ \blacktriangle \simeq I_{\mathbf{C}}$$

so with two natural transformations (3.15) from Corollary 7,

$$\tau = (T_e\sigma) \bullet (\varphi^{-1} F_{\mathrm{st}}) \quad \text{and} \quad \tau^{-1} = (\varphi S_{\mathrm{nd}}) \bullet (T_e\sigma^{-1}),$$

and hence we need only to show that their vertical composition is $\tau^{-1} \bullet \tau = \psi$, which is the necessary condition (3.35) of Definition 14 that this category is symmetry-extended as well.

We have shown previously that each arrow of an imploded category must be an identity or an isomorphic arrow. So, for every object $J(f)$ of $\mathbf{C} \downarrow \mathbf{C}$ obtained from the arrow $f : a \to b$ of $\mathbf{C}$, we have that

$$\begin{aligned}\tau(J(f)) &= ((T_e\sigma) \bullet (\varphi^{-1}F_{\text{st}}))(J(f)) \\ &= (T_e\sigma)(J(f)) \circ \varphi^{-1}(a) \\ &= T_e(\text{id}_a; f) \circ \varphi^{-1}(a)\end{aligned}$$

where $\varphi^{-1}(a) : a \simeq \widetilde{\text{id}_a}$ is the isomorphic arrow component of the natural isomorphism, and other arrow $T_e(\text{id}_a; f) : \widetilde{\text{id}_a} \to \widetilde{f}$ must be an isomorphic arrow (because it is not generally an identity arrow), and hence $\tau(J(f)) : a \to \widetilde{f}$ as composition of two isomorphic arrows must be an isomorphic arrow as well, and hence from the commutative diagram (3.56), we obtain

$$\tau(J(f)) = \text{is}_1 : a \to \widetilde{f}. \tag{3.59}$$

Analogously,

$$\begin{aligned}\tau^{-1}(J(f)) &= ((\varphi S_{\text{nd}}) \bullet (T_e\sigma^{-1}))(J(f)) \\ &= \varphi(b) \circ (T_e\sigma^{-1})(J(f)) \\ &= \varphi(b) \circ T_e(f; \text{id}_b)\end{aligned}$$

where $\varphi(b) : \widetilde{\text{id}_b} \simeq b$ is the isomorphic arrow component of the natural isomorphism, and other arrow $T_e(f; \text{id}_b) : \widetilde{f} \to \widetilde{\text{id}_b}$ must be an isomorphic arrow (because it is not generally an identity arrow), and hence $\tau^{-1}(J(f)) : \widetilde{f} \to b$ as composition of two isomorphic arrows must be an isomorphic arrow as well, and hence from the commutative diagram (3.56), we obtain

$$\tau^{-1}(J(f)) = \text{is}_2^{-1} : \widetilde{f} \to b. \tag{3.60}$$

Thus, we obtain that

$$\begin{aligned}(\tau^{-1} \bullet \tau)(J(f)) &= \tau^{-1}(J(f)) \circ \tau(J(f)) \\ &= \text{is}_2^{-1} \circ \text{is}_1 \quad \text{from (3.59) and (3.60)} \\ &= f \quad \text{from the commutative diagram (3.56)} \\ &= \psi(J(f)).\end{aligned}$$

Thus, from the fact that this is valid for every object of $\mathbf{C} \downarrow \mathbf{C}$, we obtain that

$$\tau^{-1} \bullet \tau = \psi : F_{\text{st}} \xrightarrow{\cdot} S_{\text{nd}},$$

and hence, Definition 14 for a symmetry-extended category is satisfied. □

Thus, any base imploded category $\mathbf{C}$ is equivalent to each of its n-dimensional levels $\mathbf{C}_n$, for $n \geq 1$, with their comma-induced adjunctions,

$$(Te_n, \blacktriangle_n, \varphi_n, \eta_n) : \mathbf{C}_{n+1} \to \mathbf{C}_n,$$

and by their composed adjunctions between

$$T_e \circ Te_2 \cdots \circ Te_n : \mathbf{C}_{n+1} \to \mathbf{C} \quad \text{and} \quad \blacktriangle \circ \blacktriangle_2 \cdots \circ \blacktriangle_n : \mathbf{C} \to \mathbf{C}_{n+1}.$$

Remark. In order to understand immediately how the structural properties of the limit case of imploded categories are poor, consider that, from the fact that they are also symmetry-extended categories, all arrows of the commutative diagram (3.47) are the isomorphisms. Consequently, the "minimal" imploded categories are the most simple (structurally) *discrete categories*. In fact, the skeleton of every imploded category is just a discrete category, where there is exactly the *bijection* between the arrows and the objects of a category. □

We recall that all n-dimensional levels $\mathbf{C}_n$, for $n \geq 2$, of a given imploded category $\mathbf{C}$ are equivalent to this base category. This fact explains why the imploded categories represent the last level in the hierarchy of internal symmetry of the categories, and we cannot obtain another subclass of the conceptually symmetric categories by addition of the new constraints.

So, finally, we obtained the following ISA (inclusive) hierarchy between different classes of conceptually symmetric categories between their primitive concepts (objects and arrows):

Symmetry	Subclass	Added operations/constraints
No	Categories	no constraints
Yes	Perfectly symmetric	Operation B_{T} from arrows into objects, with $B_{\mathsf{T}}(g \circ f) = B_{\mathsf{T}}(g) * B_{\mathsf{T}}(f)$, $B_{\mathsf{T}}(\mathrm{id}_a) \simeq a$
Yes	Conceptually closed (CoCC)	Closure $T_e : \mathbf{C} \downarrow \mathbf{C} \to \mathbf{C}$, with $T_e^0 = B_{\mathsf{T}}\psi$
Yes	Symmetry-extended	$\tau^{-1} \bullet \tau = \psi$
Yes	Imploded	$\blacktriangle \circ T_e \simeq I_{\mathbf{C}\downarrow\mathbf{C}}$

4 Internal symmetry and logical deduction

4.1 Natural deduction system

Natural deduction in its modern form was independently proposed by the German mathematician, Gerhard Gentzen in 1934, in a dissertation delivered to the faculty of mathematical sciences of the University of Göttingen. Gentzen was motivated by a desire to establish the consistency of number theory. He was unable to prove the main result required for the consistency result, the cut elimination theorem—the Hauptsatz—directly for natural deduction. For this reason, he introduced his alternative system, the sequent calculus, for which he proved the Hauptsatz both for classical and intuitionistic logic. In a series of seminars in 1961 and 1962, Prawitz gave a comprehensive summary of natural deduction calculi, and transported much of Gentzen's work with sequent calculi into the natural deduction framework.

Natural deduction grew out of a context of dissatisfaction with the axiomatizations of deductive reasoning common to the systems of Hilbert, Frege and Russell (see, e. g., Hilbert system), axioms from (1) to (9) of the Classic Propositional Logic (CPL) provided in Section A.1 in the Appendix.

In the natural deduction system of Gentzen and Prawitz, a deduction consists in deriving a proposition (formula, denoted by $a, b, c, d, \ldots$ in what follows, while logical conjunction is denoted by product '$\times$', disjunction by sum '+', while the implication by '$\Rightarrow$') from a finite number of packets of assumptions, using some predefined inference rules. Technically, packets are multisets of propositions. During the course of a deduction, certain packets of assumptions can be "closed" or "discharged."

A *proof* is a deduction such that all the assumptions have been discharged. In order to formalize the concepts of deduction, one faces the problem of describing rigorously the process of discharging packets of assumption. The difficulty is that one is allowed to discharge any number of occurrences of the same proposition in a single step, and this requires some form of tagging mechanism.

4.1.1 First solution for tagging techniques of natural deduction

The first solution

This one, used by Gentzen and Prawitz [18], consisting in viewing a deduction as a tree whose nodes are labeled with propositions. One is allowed to tag any set of occurrences of the same proposition with a natural number, which tags the inference that triggers the simultaneous discharge of all the occurrences tagged by that number. Now we can introduce the set of axioms and inference of natural deduction, as follows:

– The axioms and inference rules of the system Σ_M (*minimal positive propositional logic*) are listed below (a formula written within square brackets above premises

https://doi.org/10.1515/9783111081670-004

is to indicate that assumptions of this form occurring above the premises are discharged at this inference).

1. The axiom:

$$\frac{a}{a}\quad (\mathrm{R})$$

2. The inference rules for logical connectives (introduction and elimination):

$$\frac{\overset{[a]}{b}}{a \Rightarrow b}\quad (\rightarrow\Rightarrow), \qquad \frac{a \Rightarrow b\,,\ a}{b}\quad (\Rightarrow\rightarrow)$$

$$\frac{a\ ,\ b}{a \times b}\quad (\rightarrow \times)$$

$$\frac{a \times b}{a}\quad (\times \rightarrow)_1, \qquad \frac{a \times b}{b}\quad (\times \rightarrow)_2$$

$$\frac{a}{a + b}\quad (\rightarrow +)_1, \qquad \frac{b}{a + b}\quad (\rightarrow +)_2$$

$$\frac{a + b \quad \overset{[a]}{d} \quad \overset{[b]}{d}}{d}\quad (+ \rightarrow)$$

– The axioms and inference rules of the system Σ_I (*positive intuitionistic propositional logic*) are obtained by the system Σ_M adding the following intuitionistic absurdity rule (the constant $\perp$ is introduced for absurdity or *falsehood*):

$$\frac{\perp}{a}\quad (\perp \rightarrow) \tag{4.1}$$

The inferences are broken into atomic steps in such a way that each step involves only one logical connective. The steps are of two kinds, and for each logical connective there are inferences of both kinds: steps that allow the *introduction* of logical connectives ($(\rightarrow \times)$, $(\rightarrow\ +)$ and $(\rightarrow\Rightarrow)$), and steps that allow the *elimination* of the logical connective, $((\times \rightarrow)$, $(+ \rightarrow)$ and $(\rightarrow\Rightarrow))$.

Example 12. Let us consider the following example of the natural deduction tree:

$$\dfrac{\dfrac{\dfrac{(a \Rightarrow (b \Rightarrow c))^{\,3}\ ,\ a^{\,1}}{b \Rightarrow c} \qquad \dfrac{a^{\,1}\ ,\ (a \Rightarrow b)^{\,2}}{b}}{\dfrac{\dfrac{c}{(a \Rightarrow c)^{\,1}}}{((a \Rightarrow b) \Rightarrow (a \Rightarrow c))^{\,2}}}}{(a \Rightarrow (b \Rightarrow c)) \Rightarrow ((a \Rightarrow b) \Rightarrow (a \Rightarrow c))^{\,3}}$$

We are particularly interested in the inference rule for the introduction of the implication ($\rightarrow\Rightarrow$), based (as a rule of disjunction elimination ($+\ \rightarrow$)) on the hypothetical derivation. Each *hypothetical derivation* has a collection of antecedent derivations written on the top line (by $[a]$ in this case), and a succedent judgment written on the bottom line (judgment b in our case). The notion of hypothetical judgment *is internalized* as the connective of the implication. This fact of "internalized," in our categorial framework corresponds to the derivation of this implication in a base category from an arrow $f : a \rightarrow b$ of this category in which the arrows corresponds to logical derivations.

Let us denote by **ND** this category for the natural deduction of the propositional formula from other *true* formula. In this category, we have two kinds of arrows, deductive and non-deductive, and two kinds of objects, logical formulae and nonlogical formulae (having tensor products as connectives as well). It means that each inference rule must correspond to an arrow (deduction) in this category, from the upper formula to the derived formula, and hence there are the following properties that this category **ND** has to satisfy:

1. The objects of **ND** are initially defined *true formulae* in a set Γ of assumptions, and also all other derived objects from them (true formulae derived by inference rules, products, tensor products, etc.).
2. Between two objects of **ND** can be at most *one arrow* (derivation) as in partial orders. The derivation $\frac{a}{a}$ given by axiom (R) of a proposition a corresponds to the unique identity *deductive* arrow $\mathrm{id}_a : a \rightarrow a$ of this object a. The deductive derivation $\frac{[a]}{b}$ of the logic formula b from a logic formula a, used in the inference rules of natural deductions, is represented by the *deductive* arrow $f : a \rightarrow b$ in **ND**.
3. The structural operation "comma" used in inference rules ($\Rightarrow\rightarrow$) and ($\rightarrow \times$), which represents the set of two formulae, will be represented by the tensor product operator $\otimes$. Consequently, any *pair* of two formulae, which *is not a logical formula*, is represented by the single object $a \otimes b$ (which *is not* a limit object in **ND**), respectively. So, the inference rule ($\rightarrow \times$) is represented in **ND** by the following *nondeductive* arrow (because the tensor product $a \otimes b$ is not a logical formula):

$$\pi : a \otimes b \rightarrow a \times b. \tag{4.2}$$

The point 3 above introduces the objects (tensor) products, which *are not* logical formulae while, from rule ($\rightarrow\Rightarrow$), the logical implication $a \Rightarrow b$ must be introduced from derivation of b from a (a *deductive* arrow in **ND** from formula a to formula b). The necessity to introduce the logical implication $a \Rightarrow b$, as an object in **ND**, from a deductive arrow (derivation) $f : a \rightarrow b$ of **ND**, means that we need the category symmetry, with the conceptual-closure functor $T_e : \mathbf{ND} \downarrow \mathbf{ND} \rightarrow \mathbf{ND}$, such that the implication is defined to be the conceptualized object of **ND**:

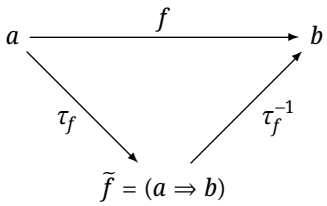
$$(a \Rightarrow b) \triangleq \widetilde{f} = T_e^0 J(f) \quad \text{iff} \quad f : a \rightarrow b \text{ is a deductive arrow.} \tag{4.3}$$

Note that $\tilde{f} = T_e^0 J(f)$ is an object of **ND** also if $f : a \to b$ is a nondeductive arrow, e. g., for the arrow π in (4.2), so that we have the nonlogical object $\tilde{\pi} = T_e^0 J(\pi)$ but we have no logical object $(a \otimes b) \Rightarrow (a \times b)$. So, we can try to reasonably use the extended-symmetry properties for this new category **ND**.

Thus, the inference rule $(\to\Rightarrow)$ means that, if hypothetically we have the deduction (a deductive arrow $f : a \to b$ in **ND**), then we must have the implication $a \Rightarrow b$ *iff* both a and b are logical formulae. So, this inference rule $(\to\Rightarrow)$ can be represented in **ND** by the following nondeductive isomorphic arrow:

$$\mathrm{is}_{\Rightarrow} : b \to (a \Rightarrow b) \tag{4.4}$$

with its inverse arrow $\mathrm{is}_{\Rightarrow}^{-1} : (a \Rightarrow b) \to b$, such that $\mathrm{is}_{\Rightarrow}^{-1} \circ \mathrm{is}_{\Rightarrow} = \mathrm{id}_b$ is the identity arrow of the object (derived formula from this inference rule) b. So, we obtain the following commutative diagram for each arrow $f : a \to b$:



(4.5)

where $\tau_f \triangleq \mathrm{is}_{\Rightarrow} \circ f$ and $\tau_f^{-1} \triangleq \mathrm{is}^{-1}$, i. e., **ND** has to be symmetry-extended category.

We can consider now the rest of the inference rules. The two elimination rules for the logical conjunction $\times$, $(\times \to)_1$ and $(\times \to)_2$ can be directly represented by the first and second *deductive* projections (epimorphisms) $\pi_1 : a \times b \to a$ and $\pi_2 : a \times b \to b$ relatively in **ND**, by considering the objects $a \times b$ as products (limits) in **ND**.

From the fact that from an object a into b in **ND**, we can have at most one arrow $f : a \to b$, in the case when both a and b are logical formulae and f is a deductive arrow, the conceptualized object $a \Rightarrow b$ represents the derivation of b from a, and hence the products $(a \Rightarrow b) \times a$ represent the application of this derivation to the formula a in order to obtain the derived formula b. That is, we obtain the unique "evaluation" *deductive* arrow $\mathrm{eval}_{a,b} : (a \Rightarrow b) \times a \to b$ in **ND**, as a composition of two deductive arrows:

$$\mathrm{eval}_{a,b} \triangleq f \circ \pi_2 : (a \Rightarrow b) \times a \to b. \tag{4.6}$$

The introduction rule for conjunction $a \times b$ can be interpreted by the standard bifunctor $\times$: **ND** $\times$ **ND** $\to$ **ND**, and hence by request that **ND** for any two objects a and b must have the *limit* (product) $a \times b$. This product *is a logical conjunction iff both objects a and b are logical formulae*. In this way, also the "modus ponens" rule of elimination of implication $(\Rightarrow\to)$ can be represented by using the bifunctor $\times$: **ND** $\times$ **ND** $\to$ **ND** in order to transform the pair $(a \Rightarrow b, a)$ into the product (limit object) $(a \Rightarrow b) \times a$.

From the fact that the domain of this evaluation arrow, for a given arrow $f : a \to b$, i. e., for the object $J(f)$ in the arrow category **ND** $\downarrow$ **ND**, is equal to

$$\begin{aligned}(a \Rightarrow b) \times a &= T_e^0(J(f)) \times F_{\text{st}}^0(J(f)) \\ &= \times(T_e^0(J(f)), F_{\text{st}}^0(J(f))) \\ &= \times\langle T_e, F_{\text{st}}\rangle(J(f))\end{aligned}$$

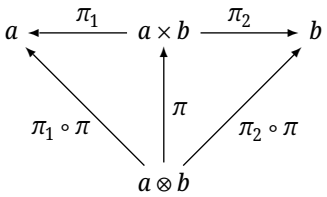
for the composed functor $\langle T_e, F_{\text{st}}\rangle$: **ND** $\downarrow$ **ND** $\to$ **ND** $\times$ **ND**, and the bifunctor $\times$: **ND** $\times$ **ND** $\to$ **ND**, we can define the new descending functor,

$$A_{pp} \triangleq \times \circ \langle T_e, F_{\text{st}}\rangle : \mathbf{ND} \downarrow \mathbf{ND} \to \mathbf{ND}. \tag{4.7}$$

So, the "modus ponens" inference rule ($\Rightarrow\!\to$), by using the tensor product $\otimes$: **ND**$\times$**ND** $\to$ **ND** for the structural comma operation, can be given by the unique composed arrow

$$m_p \triangleq \text{eval}_{a,b} \circ \pi : (a \Rightarrow b) \otimes a \to b. \tag{4.8}$$

Note that, differently from the product $\times$, which generates the limit cones with deductive projections π_1 and π_2, the tensor product $\otimes$ does not generate the limit cones, but has the two projections $\pi_1 \circ \pi$ and $\pi_2 \circ \pi$ as well, from the following commutative diagram valid for *all* objects with nondeductive arrow π:

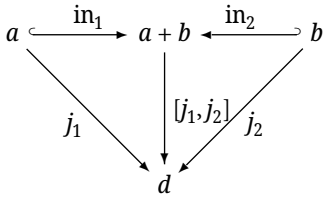


(4.9)

The consequence of this fact (that tensor product is not a limit) is that we have no inverse arrow to the arrow π, from $a \times b$, which is conjunctive logic formula (iff both a and b are logical formulae) to $a \otimes b$, which interprets the ordered *set* of two formulae a and b.

So, it remains only to find the solution for the inference rules with logical disjunction +: the rules of introduction $(\to +)_1$ and $(\to +)_2$ can be directly represented as the arrow components (monomorphisms) of the cocone of the colimit (coproduct) $a + b$, i. e., deductive injections $\text{in}_1 : a \hookrightarrow a + b$ and $\text{in}_2 : b \hookrightarrow a + b$, respectively.

Then the last rule of disjunction elimination $(+ \to)$, with the two hypothetical judgments, can be expressed by the statement that: if there exists another cocone with the vertex d and with two arrow components of this cocone $j_1 : a \to d$ and $j_2 : b \to d$ (corresponding to "hypothesis"), then there is a unique arrow $\langle j_1, j_2\rangle : a + b \to d$, with



(4.10)

Coproduct $a + b$ is *a logical conjunction formula iff both objects a and b are logical formulae*. Consequently, the three inference rules with logical disjunction + require only that **ND** have the coproducts. With this, we finished the analysis of the first solution of tagging techniques, and hence we can consider the second one.

4.1.2 Sequent-based solution for tagging techniques

The second solution consists in keeping a record of all undischarged assumptions at every stage of the deduction. We use variables for the labels, and a packet labeled with x, consisting of occurrences of the proposition a, is written as $x : a$. Thus, a deduction is a tree whose nodes are labeled with expressions of the form $x_1 : a_1, \ldots, x_n : a_n \vdash b$, with $x_i \neq x_k$ for all $k \neq i$, called sequents, where b is a proposition, and $\Gamma = x_1 : a_1, \ldots, x_n : a_n$ is a list of all undischarged assumptions at the stage of the deduction associated with this node. It is convenient to tag packets of assumptions with labels, in order to discharge the propositions in these packets in a single step. The axioms and inference rules of the system Σ_M (*minimal positive propositional logic*), in this sequent-based formulation, are listed below.

1. The axiom:

$$\Gamma, x : a \vdash a \quad \text{(R)}$$

2. The inference rules for logical connectives (introduction and elimination):

$$\frac{\Gamma, x : a \vdash b}{\Gamma \vdash a \Rightarrow b} \ (\rightarrow\Rightarrow), \qquad \frac{\Gamma \vdash a \Rightarrow b \quad \Gamma \vdash a}{\Gamma \vdash b} \ (\Rightarrow\rightarrow)$$

$$\frac{\Gamma \vdash a \quad \Gamma \vdash b}{\Gamma \vdash a \times b} \ (\rightarrow \times)$$

$$\frac{\Gamma \vdash a \times b}{\Gamma \vdash a} \ (\times \rightarrow)_1, \qquad \frac{\Gamma \vdash a \times b}{\Gamma \vdash b} \ (\times \rightarrow)_2$$

$$\frac{\Gamma \vdash a}{\Gamma \vdash a + b} \ (\rightarrow +)_1, \qquad \frac{\Gamma \vdash b}{\Gamma \vdash a + b} \ (\rightarrow +)_2$$

$$\frac{\Gamma \vdash a + b \quad \Gamma, x : a \vdash d \quad \Gamma, y : b \vdash d}{\Gamma \vdash d} \ (+ \rightarrow)$$

Let us consider the Example 12 of natural deduction by this sequent calculus:

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash a \Rightarrow (b \Rightarrow c) \quad \Gamma \vdash a}{\Gamma \vdash b \Rightarrow c} \qquad \dfrac{\Gamma \vdash a \quad \Gamma \vdash a \Rightarrow b}{\Gamma \vdash b}}{x_3 : a \Rightarrow (b \Rightarrow c),\ x_2 : a \Rightarrow b,\ x_1 : a \vdash c}}{x_3 : a \Rightarrow (b \Rightarrow c),\ x_2 : a \Rightarrow b \vdash a \Rightarrow c}}{x_3 : a \Rightarrow (b \Rightarrow c) \vdash (a \Rightarrow b) \Rightarrow (a \Rightarrow c)}}{\vdash\ (a \Rightarrow (b \Rightarrow c)) \Rightarrow ((a \Rightarrow b) \Rightarrow (a \Rightarrow c))}$$

Remark. Next, a bold letter, like $\mathbf{c}$, denotes an object of **ND**, which *is not logical formula*, while nonbold letters denote the logical formulae. □

The first step, in order to translate this sequent calculus into categorial language, is to define the categorial interpretation of it.

Definition 16 (Categorial interpretation of sequents). The interpretation by the categorial language is based on the representation of the sequent by an arrow, and the assumptions $\Gamma = x_1 : a_1, \ldots, x_n : a_n$ by the tensorial product $\mathbf{c} = a_1 \otimes \cdots \otimes a_n$ nominate by the "context object," while the sequents with the composed left-hand side $\Gamma, x : a \vdash b$ is interpreted by the next two arrows:

1. The deduction-sequent arrow $f : \mathbf{c} \otimes a \to b$.
2. A precondition arrow $g : a \to b$, used to derive the deduction arrow f.

In this case, from the fact that in **ND** we have, from the commutative diagram (4.9), the projection tensor product arrow $\pi_2 \circ \pi : \mathbf{c} \otimes a \to a$, by using this precondition arrow g, we obtain the following composition of the deduction-sequent arrow:

$$f = g \circ \pi_2 \circ \pi : \mathbf{c} \otimes a \to b. \tag{4.11}$$

With this categorial interpretation of the sequents in the **ND** category, we obtain the following transformation of the sequent system above in the categorial language:

1. Axiom (R) with precondition arrow $\mathrm{id}_a : a \to a$ (identity arrow of the object a always there exists in a category), is represented by the arrow $f = \mathrm{id}_a \circ \pi_2 \circ \pi = \pi_2 \circ \pi : \mathbf{c} \otimes a \to a$.
2. The inference rule for introduction of implication ($\to\Rightarrow$), for the precondition arrow $f : a \to b$ of **ND**, which defines the implication object $(a \Rightarrow b) = T_e^0(J(f))$ and the arrow $k = f \circ \pi_2 : \mathbf{c} \times a \to b$, is traduced by

$$\frac{k \circ \pi : \mathbf{c} \otimes a \to b}{\Lambda(k) : \mathbf{c} \to a \Rightarrow b} \quad (\to\Rightarrow)$$

where from (4.2) we have the arrow $\pi : \mathbf{c} \otimes a \to \mathbf{c} \times a$, and we obtain the constraint

$$\text{if there exists an arrow } k = f \circ \pi_2 : \mathbf{c} \times a \to b, \quad \text{then must exist } \Lambda(k) : \mathbf{c} \to a \Rightarrow b \tag{4.12}$$

where $\Lambda(k)$ is the "abstraction" of the arrow k.

3. The inference rule for elimination of implication $(\Rightarrow\to)$ is traduced by

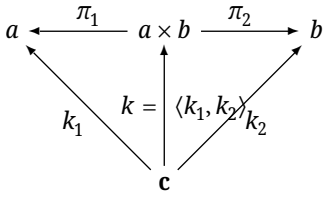
$$\frac{h : \mathbf{c} \to a \Rightarrow b \quad k : \mathbf{c} \to a}{f \circ k : \mathbf{c} \to b} \quad (\Rightarrow\to)$$

where $f : a \to b$ is an arrow between two logical formulae, for which there exists the object (logical implication) $(a \Rightarrow b) = T_e^0(J(f))$ in **ND**.

4. The inference rule for introduction of product $(\to \times)$ is traduced by

$$\frac{k_1 : \mathbf{c} \to a \quad k_2 : \mathbf{c} \to b}{\langle k_1, k_2 \rangle : \mathbf{c} \to a \times b} \quad (\to \times)$$

as it follows from the commutative diagram for the product (limit) cone:



(4.13)

5. The inference rules for elimination of product $(\times \to)$ are traduced by (see the commutative diagram above)

$$\frac{k : \mathbf{c} \to a \times b}{\pi_1 \circ k : \mathbf{c} \to a} \quad (\times \to)_1, \qquad \frac{k : \mathbf{c} \to a \times b}{\pi_2 \circ k : \mathbf{c} \to b} \quad (\times \to)_2$$

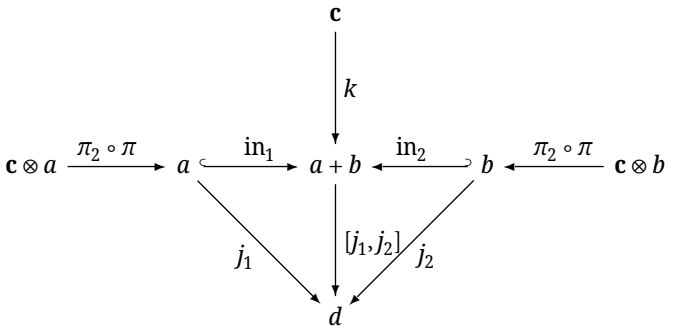
6. The inference rules for introduction of coproduct $(\to +)$ is traduced by (see the commutative diagram of coproduct cocone (4.10)),

$$\frac{k_1 : \mathbf{c} \to a}{\mathrm{in}_1 \circ k_1 : \mathbf{c} \to a + b} \quad (\to +)_1, \qquad \frac{k_2 : \mathbf{c} \to b}{\mathrm{in}_2 \circ k_2 : \mathbf{c} \to a + b} \quad (\to +)_2$$

7. The inference rule for elimination of coproduct $(+ \to)$, by Definition 16, introduces two precondition arrows: $j_1 : a \to d$ from conditional sequent $\Gamma, x : a \vdash d$ and $j_2 : b \to d$ from conditional sequent $\Gamma, y : b \vdash d$ used to derive the deduction-sequent arrows by (4.11), we obtain

$$\frac{k : \mathbf{c} \to a + b \quad j_1 \circ \pi_2 \circ \pi : \mathbf{c} \otimes a \to d \quad j_2 \circ \pi_2 \circ \pi : \mathbf{c} \otimes b \to d}{[j_1, j_2] \circ k : \mathbf{c} \to d} \quad (+ \to)$$

as follows from the extension of the cocone coproduct diagram (4.10):



(4.14)

Thus, by enabling the natural deduction also with the sequent-based tagging technique, we obtained only the new constraint provided by (4.12).

4.2 Definition of symmetry-extended category ND for natural deduction

We have introduced informally in the previous section the symmetry-extended category **ND**, such that for a given set of assumptions (set of true formulae) Γ, for each inference rule introduce an arrow, as explained previously, so that this category for each two objects would have at most one arrow between them as in partial orders.[1]

1 In what follows will be provided a technically different approach to definition of the natural deduction category **ND** w. r. t. that is used in [4]. Here, we introduce also the tensorial coproduct as a categorial interpretation of the structural "comma" connective. The ad hoc version used in my PhD thesis, where for the arrows are used a binary relation R in Definition 5, Section 3.2 of [4], so that each arrow f is technically represented by the unique pair $(\mathrm{dom}(f), \mathrm{cod}(f))$, because we can have at maximum one arrow between two arrows in this category. So, the composition of the binary relation **R** corresponds to the composition of the arrows of this category, and for any two objects with an arrow between them and the opposite arrow as well, these two objects are isomorphic. Note that the introduction of partial mappings Ap_1, Ap_2 and mapping Θ in [4], which use this binary relation **R** (as the set of morphisms) and objects of this category, was used only to specify that this category is CoCC with closure functor $T_e = (T_e^0, T_e^1) : \mathbf{ND} \downarrow \mathbf{ND} \to \mathbf{ND}$, in order to define the arrow component of this functor T_e^1. Thus, in what follows instead of the ad hoc definitions, in these partial mappings, we will (equivalently) directly introduce this functor as a necessary requirement for this category, and derived from it two natural transformations $\tau : \mathcal{F}_{\mathrm{st}} \to^{\cdot} T_e$ and $\tau^{-1} : T_e \to^{\cdot} S_{\mathrm{nd}}$, so that the previous introduction of mapping Ap_2 we replace it by using τ^{-1}, in the way that for each arrow $f : a \to b$, which defines the conceptualized object $(a \Rightarrow b) = \tilde{f} = T_e^0(J(f))$, we have that

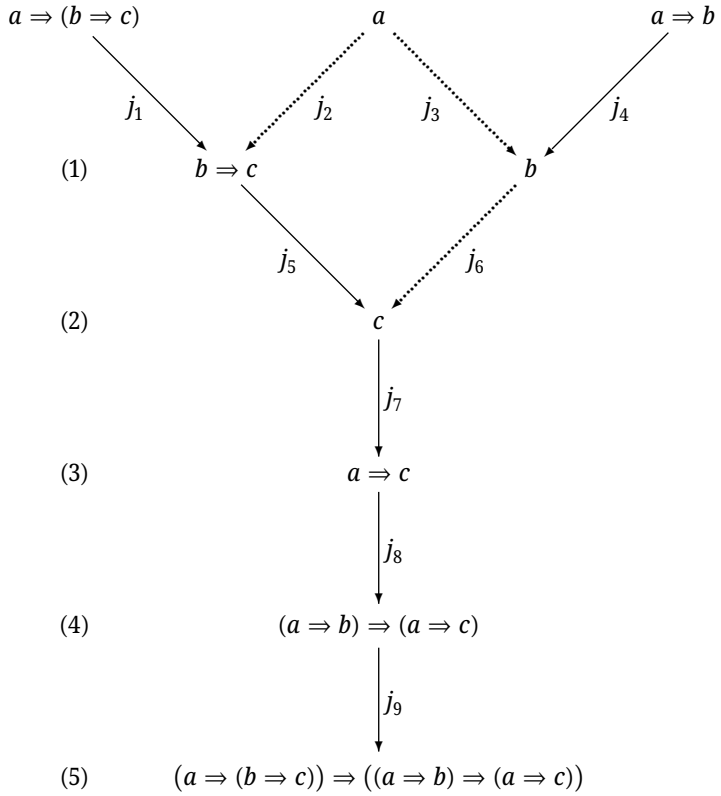
$$Ap_2(a, a \Rightarrow b) = \tau^{-1}\big(J(f)\big) : a \Rightarrow b \to b, \tag{4.15}$$

and $\Theta(f)$ is defined by its opposite arrow, required by natural deduction requirement (4.4),

$$\Theta(f) = \big(\tau^{-1}\big(J(f)\big)\big)^{-1} : b \to a \Rightarrow b, \tag{4.16}$$

so that in this more simple and direct presentation, it is enough only to set a new constraint that τ^{-1} is, from (4.4), a *natural isomorphism* in **ND**. We hope that such a technical simplification will be more intuitive and easy to follow.

Having this in mind, before the complete formal definition of this natural deduction category of Σ_M, denoted by **ND**, let us consider how the natural deduction tree in Example 12 would be represented by composition of the arrows (atomic deductions) in this category:



(4.17)

The numbers in the parenthesis denote the steps of deduction, while the dotted arrows are that from which conceptualized objects of premises (top objects in this diagram) are derived, from left to right: subformula $(b \Rightarrow c) = T_e^0(J(j_6))$, the formula (object) $(a \Rightarrow (b \Rightarrow c)) = T_e^0(J(j_2))$ and the object (formula) $(a \Rightarrow b) = T_e^0(J(j_3))$. Thus, these dotted arrows are part of the assumptions of the natural deduction expressed by diagram above. The rest of arrows are part of the deductive process.

Definition 17. For a natural deduction system with the initial set of assumptions (true formula) Γ, we define the deductive category **ND** with the following properties:

1. Γ is a strict subset of its objects.
2. For any two objects a and b, there exists *at most* one arrow $f : a \to b$ of **ND** (it is a preorder category), and their product $a \times b$ (logical conjunction iff a and b are logical formulae) with its limit cone arrows $\pi_1 : a \times b \to a$ and $\pi_2 : a \times b \to b$, their coproduct $a + b$ (logical disjunction iff a and b are logical formulae) with colimit's

cocone arrows $j_1 : a \to a + b$ and $j_2 : b \to a + b$, and finally also their tensorial product $a \otimes b$ with the arrow $\pi : a \otimes b \to a \times b$.

3. For each arrow (natural deduction) $f : a \to b$ between two logical formulae a and b, there exist the logical implication between these two formulae (objects),

$$(a \Rightarrow b) \triangleq \widetilde{f} = T_e^0 J(f) \tag{4.18}$$

obtained by the functor $T_e = (T_e^0, T_e^1) : \mathbf{ND} \downarrow \mathbf{ND} \to \mathbf{ND}$, with two derived natural transformations, $\tau : F_{\mathrm{st}} \to^{\cdot} T_e$ and *natural isomorphism* $\tau^{-1} : T_e \to^{\cdot} S_{\mathrm{nd}}$, so that in commutative diagram (4.5) we have the arrows $\tau_f = \tau(J(f))$ and $\tau_f^{-1} = \tau^{-1}(J(f))$. In correlation with such logical implication, obtained from the arrow $f : a \to b$, there exists also the arrow $\mathrm{eval}_{a,b} : (a \Rightarrow b) \times a \to b$, as specified by (4.6), and for each arrow $k = f \circ \pi_2 : c \times a \to b$ the arrow $\Lambda(k) : c \to a \Rightarrow b$ derived from (4.12).

This category is well-defined, and we have to show that it is valid for the representation principle, that is, the natural isomorphism $\varphi : T_e \circ \blacktriangle \simeq I_{\mathbf{ND}}$. That is, for each *formula* a with its identity arrow $\mathrm{id}_a : a \to a$, and hence the conceptualized object $(a \Rightarrow a) = T_e^0(J(\mathrm{id}_a))$, there is the isomorphism $a \simeq (a \Rightarrow a)$. It is easy to verify that it is true, from the existence of the arrow $\tau(J(\mathrm{id}_a)) : a \to (a \Rightarrow a)$ and its opposite arrow $\tau^{-1}(J(\mathrm{id}_a)) : (a \Rightarrow a) \to a$. So, from the fact that between two objects of **ND** there is at most one arrow, we obtain that $\tau(J(\mathrm{id}_a)) \circ \tau^{-1}(J(\mathrm{id}_a)) = \mathrm{id}_{a \Rightarrow a}$ and $\tau^{-1}(J(\mathrm{id}_a)) \circ \tau(J(\mathrm{id}_a)) = \mathrm{id}_a$, and hence, for each formula (object) a, which is by definition of **ND** a *true* formula, we obtain logically valid equivalence $a \simeq (a \Rightarrow a)$.

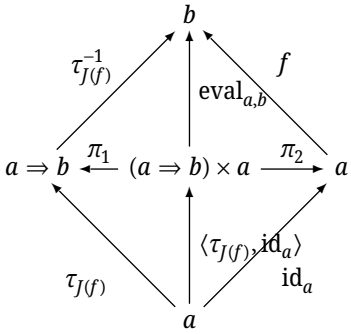
What makes Gentzen's systems especially interesting is the discovery of certain symmetry between the atomic inferences: a proof of the conclusion of an elimination is already "contained" in the proof of the premises when the major premise is inferred by introduction.

We shall refer to this by saying that the pairs of corresponding introduction and elimination satisfy the *inverse principle*: it means that nothing new is obtained by an elimination immediately following an introduction. This principle has been provided for category **ND** by Proposition 13 in [4], which here can be expressed by more simple mathematical expression, for each deductive arrow $f : a \to b$ between two formulae, relatively:

$$\tau^{-1}(J(f)) \circ (\tau^{-1}(J(f)))^{-1} = \mathrm{id}_b \quad \text{and} \quad (\tau^{-1}(J(f)))^{-1} \circ \tau^{-1}(J(f)) = \mathrm{id}_{a \Rightarrow b} \tag{4.19}$$

based on the fact that for **ND**, τ^{-1} is a natural isomorphism, so that the arrow $(\tau^{-1}(J(f)))^{-1}$: $b \to (a \Rightarrow b)$ is the inverse arrow of $\tau^{-1}(J(f)) : (a \Rightarrow b) \to b$ (expressed in [4] by (4.16) and (4.15), relatively). Thus, for each arrow $f : a \to b$, from Definition 17, we have the following commutative diagram (the bottom triangle is the commutative diagram of the limit cone of the product (limit) object $(a \Rightarrow b) \times a$ with the well-defined cone with

vertex object a, so that the arrow $\langle \mathrm{id}_a, \tau_{J(f)} \rangle$ is the unique arrow from this cone into the limit cone),



(4.20)

so that the "evaluation" arrow $\mathrm{eval}_{a,b} = \tau^{-1}_{J(f)} \circ \pi_1 = f \circ \pi_2$ there exists, as specified in (4.6). With this fact, all requirements by the natural deduction are satisfied by the category **ND**.

Now we are ready to demonstrate the follow proposition.

Corollary 9. *Each category* **ND** *is a symmetry-extended category.*

The covariant implication functor $T_e = (T^0_e, T^1_e) : \mathbf{ND} \downarrow \mathbf{ND} \to \mathbf{ND}$ *derive logic implications of two formulae by its object component* T^0_e *as conceptualized objects obtained from natural derivations (arrows of* **ND***), while its arrow component* T^1_e*, for each arrow* $(h; k) : J(f) \to J(g)$*, satisfies*

$$T^1_e(h; k) = \left(\tau^{-1}_{J(g)}\right)^{-1} \circ k \circ \tau^{-1}_{J(f)}. \tag{4.21}$$

Proof. From the commutativity of the external arrows of the diagram (4.20), we have that $\psi(J(f)) = f = \mathrm{id}_a \circ f = \tau^{-1}(J(f)) \circ \tau(J(f)) = (\tau^{-1} \bullet \tau)(J(f))$, and hence we obtain that

$$\tau^{-1} \bullet \tau = \psi$$

as required for the symmetry-extended categories.

It is easy to verify that for each arrow $(h; k) : J(f) \to J(g)$, the following diagram in **ND** commutes:

$$\begin{array}{ccc} \widetilde{f} = T^0_e(J(f)) & \xrightarrow{\tau^{-1}_{J(f)}} & S^0_{\mathrm{nd}}(J(f)) \\ \Big\downarrow T^1_e(h; k) & & \Big\downarrow k = S^1_{\mathrm{nd}}(h; k) \\ \widetilde{g} = T^0_e(J(g)) & \xleftarrow{(\tau^{-1}_{J(g)})^{-1}} & S^0_{\mathrm{nd}}(J(g)) \end{array}$$

i. e., the arrow component of the closure functor T_e is well-defined and satisfies, for each arrow $(h;k) : J(f) \to J(g)$ the equation (4.21), from the fact that from the object $\widetilde{f}$ into the object $\widetilde{g}$ in **ND** can exist *at most one* arrow. □

The more frequent interpretations for the logical implications are one of these two kinds:

– *Classical*: In classical logic, the meaning of the logic implication is defined by its truth-function, analogously as it is provided for the logical conjunction and disjunction. For this classical logic interpretation, the categorial model is defined by a (noncovariant) bifunctor $\Rightarrow$: $\mathbf{C}^{\mathrm{OP}} \times \mathbf{C} \to \mathbf{C}$, contravariant for the first and covariant for the second argument (see, e. g., in [19]), such that $(a \Rightarrow b) \triangleq (\neg a + b)$, i. e., the implication is by other two logical connectives: classical negation $\neg$ and disjunction $+$.
– *Constructive*: In nonclassical logic, an implication $a \Rightarrow b$ is defined independently from other logical connectives as the constructive approach in which a construction of formula b from a allows the inference of $a \Rightarrow b$. Historically, the intuitionistic philosophy (I. E. Brower at the beginning of last century) seams to have been the first one to realize that, by rejection of nonconstructive arguments such as, e. g., the "excluded middle" law (as provided with axiomatic system of propositional logic in Section A.1 in the Appendix).

A systematic account of the constructive interpretation of the logical connective $\Rightarrow$, along the lines of the explanations by Heyting (1956), can be given by defining inductively when something is a constitution of a sentence over a Post system S, where all inference rules are of the form $a_1, \ldots, a_n \to b$ and all a_i, for $1 \leq i \leq n$ and b are atomic formulae:

1. When a is atomic proposition, a construction $\mathbf{k}$ is simply a construction of a in S;
2. $\mathbf{k}$ is a construction of $a \Rightarrow b$ iff $\mathbf{k}$ is "constructive" function (recursive for the Church thesis), such that for each construction $\mathbf{k}'$ of a, the $\mathbf{k}(\mathbf{k}')$, i. e., the value of $\mathbf{k}$ for the argument $\mathbf{k}'$, is a construction of b.

Curry and Feyes (1958) presented the analogy between the logical constructions of propositions (intuitionistic logic) and terms (functions) of typed λ-calculus, extended by Howard (1969) and by Martin-Löf (1969) for a predicate logic: this correspondence, also known by the suggestive name of Curry–Howard "*formulae-as-types*" analogy, has been a main methodological tool for the design of new functional languages with a powerful-type system.

The Curry–Howard correspondence works only for logical systems of *intuitionistic logic*. This restriction should be clear, since the constructive, procedural interpretation of this notion of proof was the very basis of Brower's approach to mathematics, which inspired Heyting's formalization of intuitionistic logic. The "*formulae-as-types*" has a di-

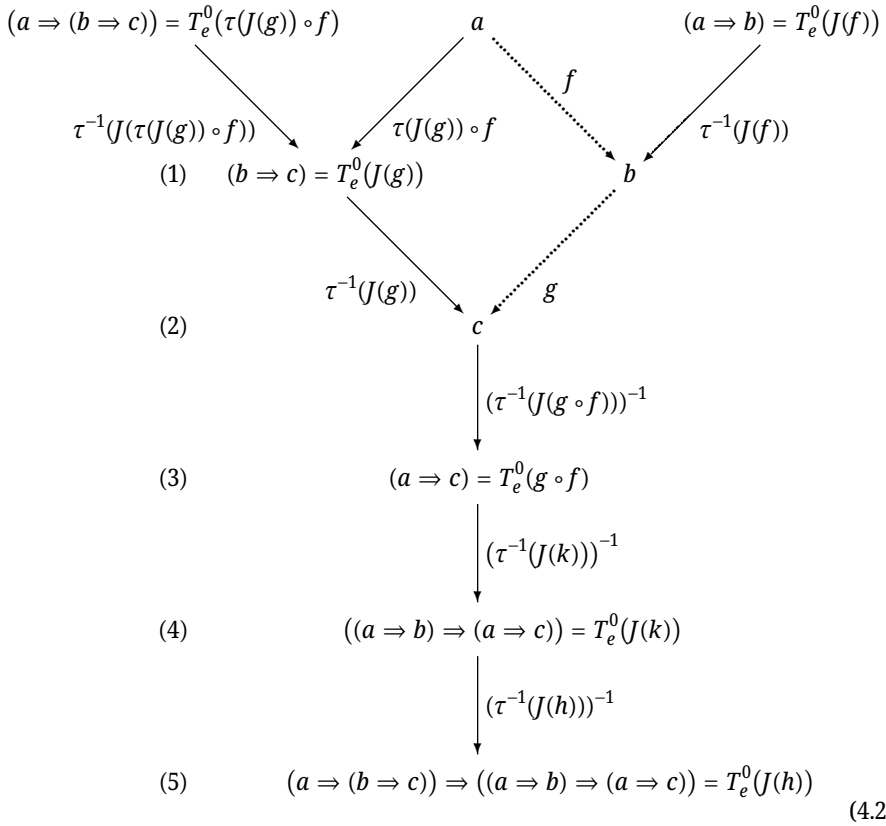
rect application to systems of natural deduction, where one proceeds by the method of drawing inferences from assumptions, considered here.

So, for the natural deduction we can introduce the new definition of the *covariant* functor for definition of the logical implication between formulae, based on the specific constructive interpretation explained above. This non-bifunctorial consideration (used for the classical logic) is a consequence of the following observations provided by the relevant logics:

1. While the logical conjunction and disjunction of two formulae a and b are binary symmetric operations (categorial products and coproducts) $a \times b = b \times a$ and $a + b = b + a$, this symmetric property does not hold for the implication. So, the implication depends of the ordering of its arguments, which is a *vectorial* property, in contraposition with the *scalar* properties of the conjunction and disjunction. Intuitively, the implication does not depend only on the truth values of its arguments, but represent a transitional property from its first into the second arguments, which is expressed, e. g., by the (possibly *relevant*) deduction of the second argument from its first argument, different from the interpretation of implication in the classical logic. This principle has been the reason to overcome this classical logic inconsistency by research of meaningful relevant logics.
2. In the actual functorial interpretation of logical connectives, there exists the adjunction between the product (and product) bifunctor and the diagonal functor $\triangle : \mathbf{C} \to \mathbf{C} \times \mathbf{C}$, while this adjunction does not exist for the exponentiation bifunctor $\Rightarrow: \mathbf{C}^{\mathrm{OP}} \times \mathbf{C} \to \mathbf{C}$ used actually for the implication of classical logic.
3. By considering the operational semantics of Gentzen (1935), provided in his natural deduction system, by the inference rule $(\to\Rightarrow)$, it is possible to infer (introduce) the implication $a \Rightarrow b$ only if there is some deduction of the formula b from a.

Consequently, the original categorial semantics for such constructive approach to the logical implication, by definition of a *covariant* functor $\Rightarrow: \mathbf{C} \downarrow \mathbf{C} \to \mathbf{C}$, has been provided [4] in my PhD thesis, and here (in a technically different way) represented by Definition 17 and Corollary 9 for application to the positive propositional fragment of natural deduction by identifying this covariant implication functor with the closure functor $T_e : \mathbf{ND} \downarrow \mathbf{ND} \to \mathbf{ND}$ provided in Definition 17.

Now, we can specify the arrows j_k, $1 \leq 9$, in the commutative diagram (4.17), based on the properties of the closure functor T_e and its natural transformations τ and τ^{-1}. We will denote by $f = j_3 : a \to b$ and $g = j_6 : b \to c$, which guarantee the existence of the conceptualized object $a \Rightarrow b = T_e^0(J(f))$ and $b \Rightarrow c = T_e^0(J(g))$, used in the assumptions of this deduction (in the top line of (4.17)), so that we obtain



(4.22)

with the composed arrow

$$
\begin{aligned}
k &= (\tau^{-1}(J(g \circ f)))^{-1} \circ g \circ \tau^{-1}(J(f)) : a \Rightarrow b \to a \Rightarrow c \quad \text{and} \\
h &= (\tau^{-1}(J(g \circ f)))^{-1} \circ g \circ \tau^{-1}(J(f)))))^{-1} \circ \\
&\quad \circ (\tau^{-1}(J(g \circ f)))^{-1} \circ \tau^{-1}(J(g)) \\
&\quad \circ \tau^{-1}(J(\tau(J(g)) \circ f)) : (a \Rightarrow (b \Rightarrow c)) \to ((a \Rightarrow b) \Rightarrow (a \Rightarrow c)).
\end{aligned}
$$

4.2.1 The properties of the covariant implication functor

So far, we have determined that the natural deduction category **ND** is a Cartesian category, where for each two objects a and b, the limit object $a \times b$ corresponding to the product there exists. What we need now is to verify if the further closure property, generally holds, or if not generally, what are the limitations.

First of all, to verify if we have the exponents of a and b, denoted by b^a, which suitably represents the set of morphisms from the object a into the object b. From the fact that in the natural deduction category **ND**, in Definition 17, it holds that at most

from a into b there exists *only one arrow*, and we conclude that if there exists such an arrow $f : a \to b$ of **ND**, then we can define the exponent object b^a to be equal (up to isomorphism) to the logical implication $(a \Rightarrow b) \triangleq T_e^0(J(f))$ obtained from the closure functor $T_e = (T_e^0, T_e^1) : \mathbf{ND} \downarrow \mathbf{ND} \to \mathbf{ND}$, so that this exponent object indeed represents the morphisms (in our case the *unique* morphism) from a into b.

From the reference to typing in programming, the key property of these conceptualized objects $a \Rightarrow b$, which represents (by symmetry property) the morphisms, provides an interpretation to a common construct in actual programming; namely, the identification of types such as $(a \times b) \Rightarrow c$ and $a \Rightarrow (b \Rightarrow c)$. This corresponds to the important uniformity property of programs of several arguments, which is directly inherited from classical recursion theory, and represented by the exponentiation commutative diagram (A.19) in Section A.2 in the Appendix.

$$\begin{array}{ccc} c^b \times b & \xrightarrow{\text{eval}_{b,c}} & c \\ {\scriptstyle \Lambda(k)}\uparrow & \nearrow_{k} & \\ a \times b & & \end{array} \qquad \text{id}_b \qquad (4.23)$$

However, **ND** has no the exponent c^b, i. e., $b \Rightarrow c$, for each pair of the objects b and c, but only if there is a deduction (an arrow) $g : b \to c$, and hence **ND** is not a Cartesian Closed Category (CCC).

Thus, in what follows, we will examine the limitations for which the exponentiation commutative diagram (4.23) above is well-defined in the natural deduction category **ND**, i. e., in some way is a "partial concept" of exponentiation.

Corollary 10. *For each product* $\mathbf{a} \times b$*, where* $\mathbf{a}$ *can be a nonformula object as well, the domain of a composed arrow* $k = g \circ \pi_2 : \mathbf{a} \times b \to c$*, with the conceptualized object* $(b \Rightarrow c) = T_e^0(J(g))$ *obtained from the deduction (arrow)* $g : b \to c$ *of the formula c from the formula b, the exponentiation diagram in* **ND** *commutes,*

$$\begin{array}{ccc} (b \Rightarrow c) \times b & \xrightarrow{\text{eval}_{b,c}} & c \\ {\scriptstyle \Lambda(k)}\uparrow & \nearrow_{k} & \\ \mathbf{a} \times b & & \end{array} \qquad \text{id}_b \qquad (4.24)$$

with the "evaluation" arrow $\text{eval}_{b,c}$ *from (*4.6*) and* $\Lambda(k)$ *from (*4.12*).*

Proof. It holds directly from the point 3 of Definition 17. In effect, having the arrow $g : b \to c$, and hence the conceptualized object (exponent) $b \Rightarrow c$ then, from (4.6), we always have the (unique) arrow $\text{eval}_{b,c} = \tau^{-1}(J(g)) \circ \pi_1 = g \circ \pi_2 : (b \Rightarrow c) \times b \to c$ of **ND**

(from the fact that from a given object into another object of **ND** we can have at most one arrow).

So, if there exists a composed arrow $k = g \circ \pi_2 : \mathbf{a} \times b \to c$ in **ND**, from the sequent-inference rule of introduction of implication, from (4.12) there exists also $\Lambda(k)$, the "abstraction" of the arrow k. The equation $k = \mathrm{eval}_{b,c} \circ (\Lambda(k) \times \mathrm{id}_b) : \mathbf{a} \times b \to c$ must be satisfied because from the object $\mathbf{a} \times b$ into the object c can exist at maximum one arrow in a category **ND**. □

Remark. Note that for the case when $\mathbf{a} = c = b$ are the same formula, we obtain $k = \pi_1 : b \times b \to b$, and hence we obtain *always*, for each formula (an object) b of **ND**, the commutative exponentiation diagram (4.24) because we have always the identity arrow $\mathrm{id}_b : b \to b$, and its conceptualized object $(b \Rightarrow b) = T^0_e(J(\mathrm{id}_b))$, and that in this case both arrows $g : b \to b$ are just the identity arrow id_b (between b and b can be at most one arrow in **ND**, and it is just the identity arrow id_b), and hence we obtain the maximal reduction $\Lambda(k) = \Lambda(\pi_1) = (\varphi(\mathrm{cod}(\pi_1)))^{-1} = (\varphi(b))^{-1} : b \to (b \Rightarrow b)$, i. e., $\Lambda(k) = \Lambda(\pi_1) = \tau(J(\mathrm{id}_b)) : b \to (b \Rightarrow b)$. □

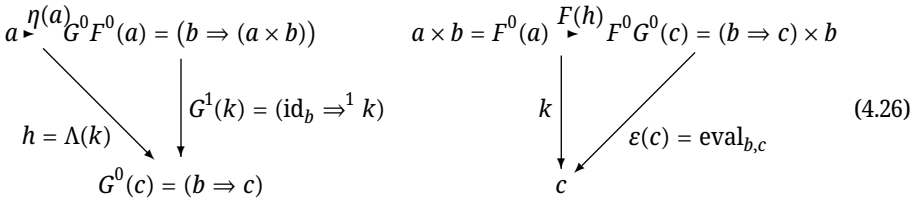
Moreover, the exponentiation diagram (4.24) can be extended to tensorial product as well:

$$\begin{array}{ccccc}
(b \Rightarrow c) \otimes b & \xrightarrow{\pi} & (b \Rightarrow c) \times b & \xrightarrow{\mathrm{eval}_{b,c}} & c \\
\uparrow{\scriptstyle \Lambda(k) \otimes \mathrm{id}_b} & & \uparrow{\scriptstyle \Lambda(k)} \quad \uparrow{\scriptstyle \mathrm{id}_b} & \nearrow{\scriptstyle k} & \\
\mathbf{a} \otimes b & \xrightarrow{\pi} & \mathbf{a} \times b & &
\end{array} \tag{4.25}$$

Corollary 10 expresses the weaker constraints for the natural deduction category **ND** w. r. t. the Cartesian Closed Categories (CCC): we do not have for each pair of objects a and b the exponent object $a \Rightarrow b$. It means that for this category **ND** we have no the adjunction between the standard diagonal functor and exponentiation bifunctor $\Rightarrow$: $\mathbf{ND}^{\mathrm{OP}} \times \mathbf{ND} \to \mathbf{ND}$ (which is usually used for the definition of CCCs). Let us now verify which kind of adjunction we can use for the definition of natural deduction category instead of the adjunction with the exponentiation bifunctor.

The transformation of the natural deduction category **ND** into the Cartesian category is easily obtained by its extension with the new tautology (or truth) constant $\top$ as the terminal object in **ND**, so that it is meaningful to analyze if **ND** can have some property of *local CCC*, from the fact that we verified that it cannot be a *global* CCC.

We recall that in actual category theory, a category **C** is a (global) CCC when it satisfies that for every object b, the (covariant) functor $F = _ \times b : \mathbf{C} \to \mathbf{C}$ has a right adjoint covariant functor $G = b \Rightarrow _ : \mathbf{C} \to \mathbf{C}$, where $\Rightarrow$: $\mathbf{C}^{\mathrm{OP}} \times \mathbf{C} \to \mathbf{C}$ is the classical exponentiation bifunctor contravariant for the first and covariant for the second argument. Consequently, this adjunction $(F, G, \varepsilon, \eta)$ is represented by the following commutative diagrams in **C**:



(4.26)

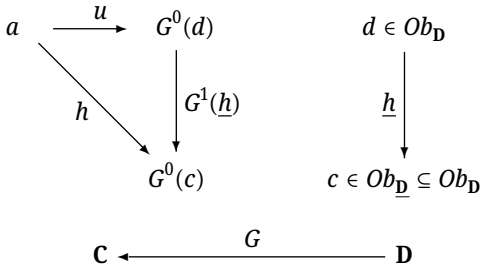
where the right-hand commutative diagram, with the arrow $F(h) = h \times \mathrm{id}_b$, is equal to exponentiation diagram (4.24).

4.2.2 Local Cartesian Closed Adjunction (pCCC)

In this section, we intend to define a kind of locally valid (partial) adjunction between the categorial products and the covariant implication functor $T_e : \mathbf{ND} \downarrow \mathbf{ND} \rightarrow \mathbf{ND}$ with the properties provided by (4.21) in Corollary 9 in a previous section.

As was previously concluded, the (global) CCC is a too strong constraint for the natural deduction Cartesian categories **ND** with the terminal object $\top$, from the fact that in them for a given two objects b and c, the "exponent object," defined as logical implication $b \Rightarrow c$, does not exist if we have no deduction (an arrow in **ND**) of the formula (an object) c from the formula b. In effect, the covariant closure functor T_e does not define the exponents $b \Rightarrow c$ if there is no an arrow $g : b \rightarrow c$ of this natural deduction category **ND**. So, at maximum we can obtain a kind of partial CCC property of **ND** (denoted by pCCC), with partial adjunctions and partial universal arrows of **ND**. Hence, we have to define formally such partial universal arrows (denominated as "p-universal arrows") as follows.

Definition 18. Let $G : \mathbf{D} \rightarrow \mathbf{C}$ be a functor and $a \in \mathrm{Ob}_{\mathbf{C}}$ an object of **C**. A p-universal arrow w. r. t. a given full subcategory $\underline{\mathbf{D}}$ of **D**, from $a \in \mathrm{Ob}_{\mathbf{C}}$ to G, is a pair (d, u) which consists of an object d of **D** and an arrow $u : a \rightarrow G^0(d)$ in **C**, such that for every pair (c, h), where c is an object of $\underline{\mathbf{D}}$ and an arrow $h : a \rightarrow G^0(c)$ in **C** there exists a unique arrow $\underline{h} : d \rightarrow c$ in **D** such that the following diagram commutes:



The *locality* of the p-arrows w. r. t. the (global) universal arrow is obtained by the restriction of the objects c to be only in the (local) subcategory $\underline{\mathbf{D}}$. In what follows, for

the natural deduction category **ND** this locality regards a prefixed object (a formula) and all deductions of other formulae (other objects) from this given object: locality in this case determines the full subcategory $\underline{\mathbf{D}} \subset \mathbf{D}$ composed by these local deductions. That is, the locality in this case corresponds to the locality of the natural deductions from a given formula.

Now we can extend this concept of locality to the adjunctions as well.

Definition 19. A partial adjunction for a covariant functor $G : \mathbf{D} \to \mathbf{C}$, local w. r. t. a full subcategory $\underline{\mathbf{D}}$ of $\mathbf{D}$, denoted by $(G, K, \varepsilon, \eta)$, with its right adjoint functor $K : \mathbf{C} \to \mathbf{D}$, is defined by the following constraints:

1. For the unit $\eta : I_{\mathbf{C}} \to^{\cdot} G \circ K$ and for each object $a \in \mathrm{Ob}_{\mathbf{C}}$ of $\mathbf{C}$, the arrow component $\eta(a) : a \to G^0K^0(a)$ is a p-universal arrow from $a \in \mathrm{Ob}_{\mathbf{C}}$ to G w. r. t. a full subcategory $\underline{\mathbf{D}}$ of $\mathbf{D}$.
2. For the counit $\varepsilon : K \circ G \to^{\cdot} I_{\mathbf{D}}$ and each object c of the subcategory $\underline{\mathbf{D}}$, the arrow component $\varepsilon(c)$ is universal from K to c.

In a special case when $\underline{\mathbf{D}} = \mathbf{D}$, a local (partial) adjunction corresponds to the global adjunction.

A partial adjunction for a covariant functor $G : \mathbf{D} \to \mathbf{C}$ can be "completed" by introducing the *disjoint partition* of the full subcategories of the category $\mathbf{D}$, as follows.

Definition 20 (Partitioned Adjunction). Let $\mathcal{H} : S_{\mathbf{D}} \to S_{\mathbf{C}}$, for the subsets of objects $S_{\mathbf{D}} \subseteq \mathrm{Ob}_{\mathbf{D}}$ and $S_{\mathbf{C}} \subseteq \mathrm{Ob}_{\mathbf{C}}$, be a partition function with the kernel $\mathrm{Ker}_{\mathcal{H}}$, such that for each object $b \in \mathrm{Im}(\mathcal{H}) \subseteq S_{\mathbf{C}}$, the equivalence class

$$[b] \triangleq \{d \in S_{\mathbf{D}} \mid b = \mathcal{H}(d)\} = \{d \mid (b, d) \in \mathrm{Ker}_{\mathcal{H}}\} \tag{4.27}$$

defines the objects of the full subcategory $\mathbf{D}_{[b]}$ of $\mathbf{D}$. Let us denote by $S_{\mathbf{D}}/\mathrm{Ker}_{\mathcal{H}}$ the set of all equivalence classes generated by the function $\mathcal{H}$.

Consequently, a partitioned adjunction (p-adjunction) for a covariant functor $G : \mathbf{D} \to \mathbf{C}$ is a pair of functions $(\mathcal{H}, \mathcal{F})$, with $\mathcal{F} : \mathrm{Ker}_{\mathcal{H}} \to \mathrm{Ob}_{\mathbf{E}}$ where $\mathbf{E} = \mathbf{D}^{\mathbf{C}}$ is the category of functors from $\mathbf{C}$ into $\mathbf{D}$, such that for each equivalence class $[b] \in \mathrm{Ker}_{\mathcal{H}}$, which determines the full subcategory $\mathbf{D}_{[b]}$ of $\mathbf{D}$, the functor $K = \mathcal{F}([b]) : \mathbf{C} \to \mathbf{D}$ is the right locally adjoint to G.

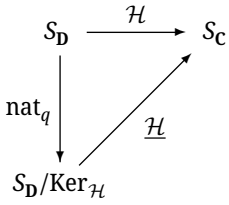
This definition of partitioned adjunction for a given covariant functor $G : \mathbf{D} \to \mathbf{C}$, which introduces the disjoint set of the full subcategories $\mathbf{D}_{[b]}$ of $\mathbf{D}$ and corresponding set of partially right adjoint "local" functors $\mathcal{F}([b]) : \mathbf{C} \to \mathbf{D}$ to G, and generalizes the standard (global) adjunction, as we can see from the following example.

Example 13. We will consider two extreme cases—the minimal one, which reduces partitioned into global adjunction, and the maximal one (with maximal possible partitions):

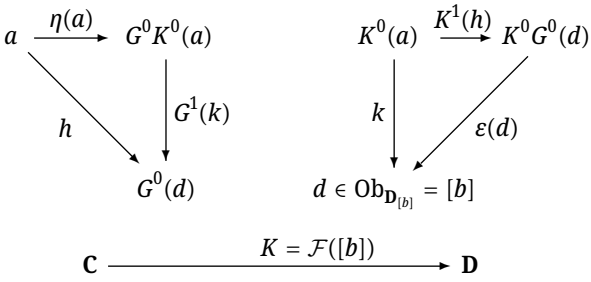
1. Minimal case: When $S_{\mathbf{D}} = \mathrm{Ob}_{\mathbf{D}}$ and $S_{\mathbf{C}} = \mathrm{Ob}_{\mathbf{C}}$, and when the partition function $\mathcal{H}$ is constant (returns the constant value b), and hence generates a unique equivalence

class $[b] = \mathrm{Ob}_{\mathbf{D}}$, which determines the full subcategory $\mathbf{D}_{[b]}$ to be equal to the category $\mathbf{D}$. Thus, we have only one functor $K = \mathcal{F}([b]) : \mathbf{C} \to \mathbf{D}$, globally right adjoint to G. Thus, the standard adjunction is this specific minimal case of the more general concept of partitioned adjunction.

2. Maximal case: When the partition function $\mathcal{H}$ is surjective for a fixed $S_{\mathbf{C}} \subseteq \mathrm{Ob}_{\mathbf{C}}$. Then $S_{\mathbf{D}}/\mathrm{Ker}_{\mathcal{H}}$ has the maximal number of the equivalence classes, equal to the number of (non isomorphic) objects of $S_{\mathbf{C}} \subseteq \mathrm{Ob}_{\mathbf{C}}$. In this case, from the theorem of decomposition, the following commutative diagram in **Set** is valid, where nat_q is a natural representation:



while $\underline{\mathcal{H}}$ is a bijective function, with its inverse $\underline{\mathcal{H}}^{-1}$. So, for each object b of $S_{\mathbf{C}}$ and equivalence class $[b] \subset S_{\mathbf{D}}$, we obtain the following local (partial) adjunction $(G, K, \varepsilon, \eta)$:



(4.28)

In the maximal case of partitioned adjunction, for the covariant functor $G = T_e$ with $\mathbf{C} = \mathbf{ND}$ and $\mathbf{D} = \mathbf{ND} \downarrow \mathbf{ND}$, we have the following theorem.

Theorem 7. *For the natural deduction category* **ND**, *there exists the following maximal-partitioned adjunction of Definition* 20 *for the covariant functor* $G = T_e : \mathbf{ND} \downarrow \mathbf{ND} \to \mathbf{ND}$, *with the pair of functions* $(\mathcal{H}, \mathcal{F})$ *specified by:*

1. $S_{\mathbf{C}} \triangleq \{a \in \mathrm{Ob}_{\mathbf{ND}} \mid a \text{ is a formula}\}$, *and* $S_{\mathbf{D}} \triangleq \{J(f) \in \mathrm{Ob}_{\mathbf{ND}\downarrow\mathbf{ND}} \mid \mathrm{dom}(f) \text{ and } \mathrm{cod}(f) \text{ are the formulae}\}$.
2. $\mathcal{H}$ *is the restriction of the function* F^0_{st} *(the object component of the first-projection functor* $F_{\mathrm{st}} : \mathbf{ND} \downarrow \mathbf{ND} \to \mathbf{ND}$*) to the subset* $S_{\mathbf{D}} \subseteq \mathrm{Ob}_{\mathbf{ND}\downarrow\mathbf{ND}}$, *and hence the surjective function.*
3. *For each equivalence class* $[b] \in \mathrm{Ker}_{\mathcal{H}}$, *the function* $\mathcal{F}$ *returns with the functor* $K = \mathcal{F}([b]) : \mathbf{ND} \to \mathbf{ND} \downarrow \mathbf{ND}$, *right p-adjoint to G, defined by*

$$\mathcal{F}([b]) \triangleq \blacktriangle \circ (_ \times b) \tag{4.29}$$

where $(_ \times b) : \mathbf{ND} \to \mathbf{ND}$ *is covariant endofunctor derived from the product bifunctor* $\times : \mathbf{ND} \times \mathbf{ND} \to \mathbf{ND}$*, and* $\blacktriangle : \mathbf{ND} \to \mathbf{ND} \downarrow \mathbf{ND}$ *is diagonal functor* (1.11)*.*

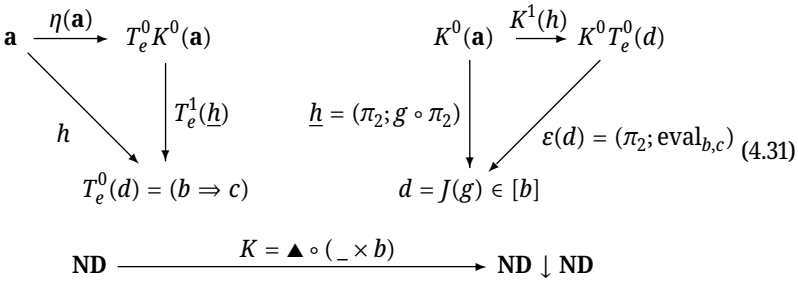
This p-adjunction represents exactly the exponentiation diagrams specified in the diagram (4.24) *of Corollary* 10*.*

Proof. The $\mathcal{H} \triangleq F^0_{\mathrm{st}} : S_{\mathbf{D}} \to S_{\mathbf{C}}$ is the surjective function from the fact that *for every* object (a formula) $b \in S_{\mathbf{C}} \subseteq \mathrm{Ob}_{\mathbf{ND}}$ with its identity arrow id_b, the object $J(\mathrm{id}_b)$ is an element of the set $S_{\mathbf{D}}$, and hence we have that $\mathcal{H}(J(\mathrm{id}_b)) = F^0_{\mathrm{st}}(J(\mathrm{id}_b)) = b$, i. e., $b \in \mathrm{Im}(\mathcal{H})$. We have that for each fixed object (which is a logical formula) $b \in S_{\mathbf{C}} \subseteq \mathrm{Ob}_{\mathbf{ND}}$, from Definition 20 and (4.27), we obtain

$$\begin{aligned} &[b] \triangleq \{d \in S_{\mathbf{D}} \mid b = \mathcal{H}(d)\} = \{J(g) \in S_{\mathbf{D}} \subseteq \mathrm{Ob}_{\mathbf{ND}\downarrow\mathbf{ND}} \mid b = F^0_{\mathrm{st}}(J(g))\}, \quad \text{i. e.,} \\ &[b] = \{J(g) \in S_{\mathbf{D}} \subseteq \mathrm{Ob}_{\mathbf{ND}\downarrow\mathbf{ND}} \mid \mathrm{dom}(g) = b\}, \end{aligned} \tag{4.30}$$

which is the set of arrows (deductions) from the object d to derived objects (formulae) in **ND**. Thus, the locality in this case is determined by the actual object b and *only* derivations of other formulae (objects) from it. So, for each object b of **ND**, we obtain that the functor (4.29), $K = \mathcal{F}([b]) \triangleq \blacktriangle \circ (_ \times b)$ is right p-adjoint to the symmetry-closure functor T_e, with p-adjunction $(G, K, \varepsilon, \eta)$ as follows from the commutative diagram (4.28) for $G = T_e$ and $d = J(g) \in [b]$, i. e., the arrow $g : b \to c$ (a deduction from b), and hence we have the logical implication $(b \Rightarrow c) = \widetilde{g} = T^0_e(J(g))$.

(We recall that the bold letter **a** represents an object of **ND**, which is not necessarily logical formulae but can be a tensorial product as well.)



(4.31)

The components of the functor K are $K^0 = \blacktriangle^0(_ \times b)$ and $K^1 = \blacktriangle^1(_ \times \mathrm{id}_b)$, so that

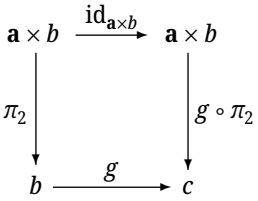
$$\begin{aligned} K^0(\mathbf{a}) &= \blacktriangle^0(\mathbf{a} \times b) = J(\mathrm{id}_{\mathbf{a}\times b}) \\ T^0_e K^0(\mathbf{a}) &= T^0_e(J(\mathrm{id}_{\mathbf{a}\times b})), \end{aligned}$$

i. e., the conceptualized object $\widetilde{\mathrm{id}}_{\mathbf{a}\times b}$ equal to the tautology $(a \times b) \Rightarrow (a \times b)$ (isomorphic to terminal object $\top$) if $\mathbf{a} = a$ is a logical formula and $\eta(a) : a \to ((a \times b) \Rightarrow (a \times b))$ is

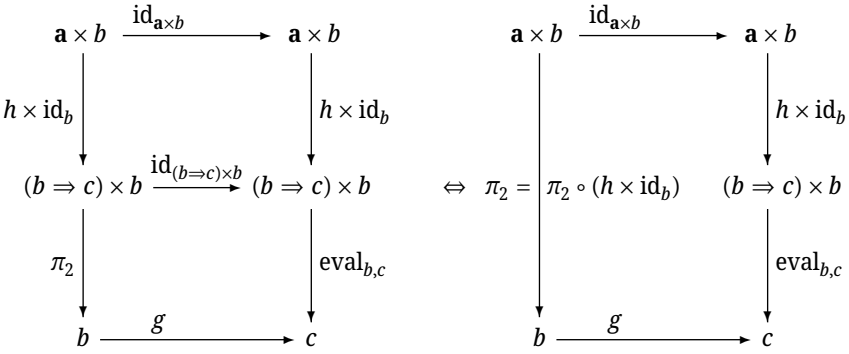
substantially unique arrow from the object a into the terminal object $\top$ of **ND**,

$$K^0 T_e^0(d) = K^0(b \Rightarrow c) = J(\mathrm{id}_{(b\Rightarrow c)\times b})$$

so that the arrow $\underline{h} : K^0(\mathbf{a}) \to d$ corresponds to the following commutative diagram in **ND**:



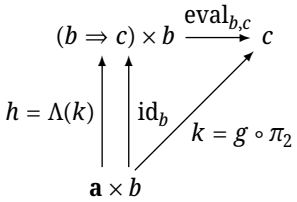
i. e., $\underline{h} = (\pi_2; g \circ \pi_2)$. Consequently, the arrow $K^1(h) = \blacktriangle^1(h \times \mathrm{id}_b) = (h \times \mathrm{id}_b; h \times \mathrm{id}_b)$ corresponds to the left-upper commutative diagram in **ND**,



while the left-hand side lower commutative diagram corresponds to the arrow $\varepsilon(d) = (\pi_2; \mathrm{eval}_{b,c})$, and from the fact that

$$\pi_2 \circ (h \times \mathrm{id}_b) = \pi_2 : \mathbf{a} \times b \to b$$

(because in **ND** between two objects, here between $\mathbf{a} \times b$ and b, can exist at most *one* arrow), the commutative diagram in the left-hand side is equivalent to the right-hand side commutative diagram, which corresponds to the exponentiation diagram (4.24) in Corollary 10,



for $h = \Lambda(k)$ (if the left-hand side diagram of p-adjunction commutes for a given h, then there exists unique arrow k such that the right-hand side diagram commutes. Thus, the abstract operator Λ is well-defined) and $k = g \circ \pi_2$ (it must be so because between $\mathbf{a} \times b$ and c can exist in **ND** at most one arrow). Moreover, from the p-adjunction in diagram (4.31): it is well-defined if there exists the pair (d, h) with the arrow $h : \mathbf{a} \to (b \Rightarrow c)$.

Hence, the exponentiation diagrams in **ND**, which must satisfy the conditions of Corollary 10, corresponds to the p-adjunction specified in this theorem. □

Consequently, the max-partitioned adjunction in this theorem (for the subset of exponents in **ND** derived by closure covariant functor T_e from the arrows (natural deductions) in **ND**) replaces the standard (global) Cartesian closed adjunction (between functors $b \Rightarrow _ : \mathbf{C} \to \mathbf{C}$ and $_ \times b : \mathbf{C} \to \mathbf{C}$, for each object b) as its weaker form fitted for the natural deduction category **ND**.

The partitioned CCC (pCCC), corresponding to these local adjunctions, can be used as models also for another applications different from natural deduction systems.

4.2.3 The *n*-dimensional levels of natural deduction categories

An arrow in the n-dimensional level $\mathbf{ND}_n$, for $n \geq 2$, generally expresses a number of correlated commutative deductive diagrams. Consequently, we want to show that also the meta-theories (the n-dimensional level $\mathbf{ND}_n$) in effect preserve all valid deductions expressed in the basic natural deduction category **ND**, only in more compressed form.

Thus, in what follows, we can demonstrate the following invariance of the structural properties of the natural deduction categories **ND** under the comma-induction transformations introduced in Section 1.3.

Theorem 8. *The structural properties of the natural deduction categories* **ND** *under the comma-induction transformations are preserved.*

That is, all n-dimensional levels $\mathbf{ND}_n = \mathbf{ND}_{n-1} \downarrow \mathbf{ND}_{n-1}$, *for* $n \geq 2$ *(with* $\mathbf{ND}_1 = \mathbf{ND}$*) preserve the structural properties of the basic natural deduction category* **ND**.

Proof. We recall that, from Proposition 9 in Section 3.4, we obtained that all n-dimensional levels $\mathbf{ND}_n$, for $n \geq 2$, are the symmetry-extended categories, "closed" by the covariant descending functor such that, for this descending functor its comma-induced family of functors, from Section 1.3, is given by

$$Te_n = \widehat{Te_{n-1}} \circ L_\perp = (JTe^1_{n-1}\psi, (Te^1_{n-1}F^1_{\mathrm{st}}; Te^1_{n-1}S^1_{\mathrm{nd}})) : \mathbf{ND}_{n+1} \to \mathbf{ND}_n \tag{4.32}$$

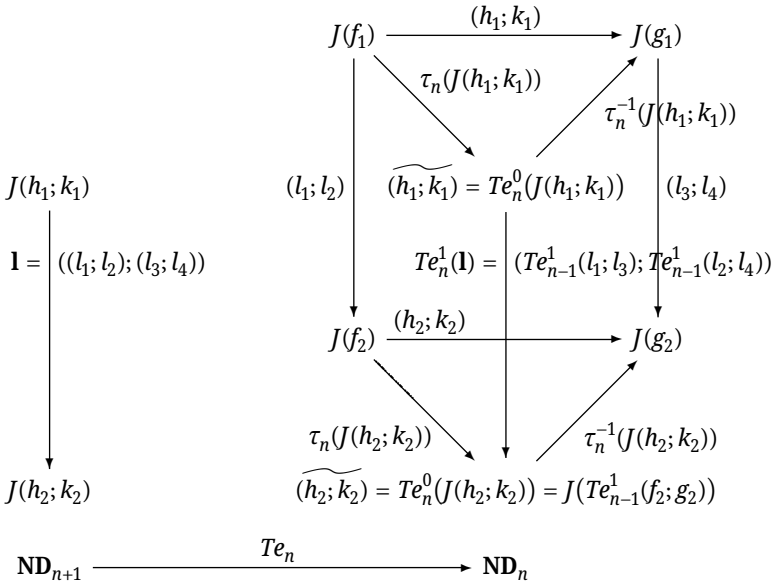
with the natural transformations

$$\begin{aligned}\tau_n &= \widehat{\tau_{n-1}L_\perp} = (\tau_{n-1}F^0_{\mathrm{st}}; \tau_{n-1}S^0_{\mathrm{nd}})L_\perp : F_{\mathrm{st}} \to Te_n \\ \tau_n^{-1} &= \widehat{\tau^{-1}_{n-1}L_\perp} = (\tau^{-1}_{n-1}F^0_{\mathrm{st}}; \tau^{-1}_{n-1}S^0_{\mathrm{nd}})L_\perp : Te_n \to S_{\mathrm{nd}}\end{aligned} \tag{4.33}$$

such that

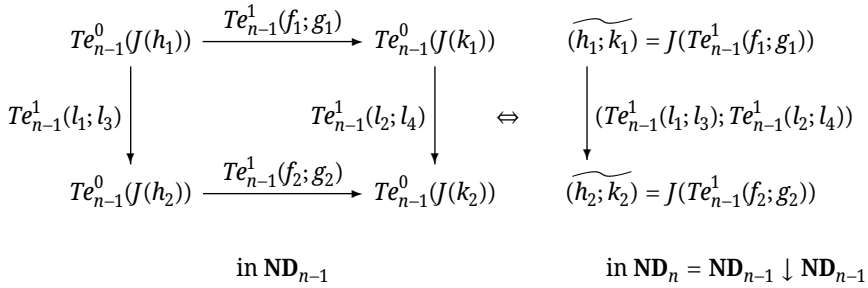
$$\tau_n^{-1} \bullet \tau_n = \psi_n. \tag{4.34}$$

Thus, from Example 11 in Section 3.5, for each arrow $\mathbf{l} = ((l_1; l_2); (l_3; l_4)) : J(h_1; k_1) \to J(h_2; k_2)$, we have the following commutative diagram, analog to (3.45):



(4.35)

where the square commutative diagram in $\mathbf{ND}_n$ corresponds to the vertical arrow in $\mathbf{ND}_{n+1}$, and the middle vertical arrow $Te^1_n(\mathbf{l}) = (Te^1_{n-1}(l_1; l_3); Te^1_{n-1}(l_2; l_4))$ in $\mathbf{ND}_n$, obtained by application of the comma-inducted functor $Te_n : \mathbf{ND}_{n+1} \to \mathbf{ND}_n$ on the vertical arrow $\mathbf{l} = ((l_1; l_2); (l_3; l_4))$ in $\mathbf{ND}_{n+1}$, and corresponds to the following commutative diagram in base category $\mathbf{ND}_{n-1}$:



with the exponents, for $i = 1, 2$,

$$(J(f_i) \Rightarrow J(g_i)) = \widetilde{(h_i; k_i)} \triangleq Te^0_n(J(h_i; k_i)) = J(Te^1_{n-1}(f_i; g_i)), \tag{4.36}$$

and the arrow components of the natural transformations,

$$\tau_n(J(h_i; k_i)) = (\tau_{n-1}(J(h_i)); \tau_{n-1}(J(k_i))), \quad \tau^{-1}_n(J(h_i; k_i)) = (\tau^{-1}_{n-1}(J(h_i)); \tau^{-1}_{n-1}(J(k_i))) \tag{4.37}$$

We have to show that for any n-dimensional level $\mathbf{ND}_n$ the requirements in points 2 and 3 of Definition 17 in Section 4.2 are satisfied (the requirement in point 1 is irrelevant because all objects of $\mathbf{ND}_n$ are determined from the arrows of the $(n-1)$-dimensional level $\mathbf{ND}_{n-1}$):

- From the fact that comma-induction is a restricted case of the more general comma-propagation, and that the comma-propagation, from Theorem 4 in Section 2.3, preserves all adjunctions, limits and colimits, it holds that the comma-induction preserves the products (limits) and coproducts (colimits), so that each n-dimensional level $\mathbf{ND}_n$ has the products and the coproducts for each pair of its objects.
- From the inductive hypothesis and the square commutative diagram (4.35) in $\mathbf{ND}_n$, the arrows, f_i, g_i, h_i, k_i, for $i = 1, 2$ and l_i, for $i = 1, 2, 3, 4$, composing the commutative diagrams in $\mathbf{ND}_{n-1}$, are unique arrows for their domain and codomain objects in $\mathbf{ND}_{n-1}$, and hence also the pairs $(h_i; k_i)$, $(l_1; l_2)$, $(l_4; l_4)$, etc., which are the arrows in $\mathbf{ND}_n$ are unique arrows between their domain and codomain objects. That is, also in the n-dimensional level $\mathbf{ND}_n$ for each pair of objects there is at most one arrow from the first into the second object.
- Also, the natural transformation τ^{-1}_n is *natural isomorphism*: in fact, its component arrows from (4.37), are $\tau^{-1}_n(J(h_i; k_i)) = (\tau^{-1}_{n-1}(J(h_i)); \tau^{-1}_{n-1}(J(k_i)))$, and from the inductive hypothesis, both its components $\tau^{-1}_{n-1}(J(h_i))$ and $\tau^{-1}_{n-1}(J(k_i))$ are isomorphic arrows of $\mathbf{ND}_{n-1}$, and hence also the arrow $\tau^{-1}_n(J(h_i; k_i))$ of $\mathbf{ND}_n$ is isomorphic.
- From (4.36), for each arrow $(h_i; k_i) : J(f_i) \to J(g_i)$ of $\mathbf{ND}_n$, for $i = 1, 2$, we obtain definition of the "meta-implication" (conceptualized) object $J(f_i) \Rightarrow J(g_i)$ of $\mathbf{ND}_n$,

$$\begin{aligned}(J(f_i) \Rightarrow J(g_i)) &= \widetilde{(h_i; k_i)} \\ &\triangleq Te^0_n(J(h_i; k_i)) \\ &= J(Te^1_{n-1}(f_i; g_i)) \\ &= J(f_i \Rightarrow g_i),\end{aligned}$$

where we denoted by $f_i \Rightarrow g_i$ the arrow $Te^1_{n-1}(f_i; g_i)$.

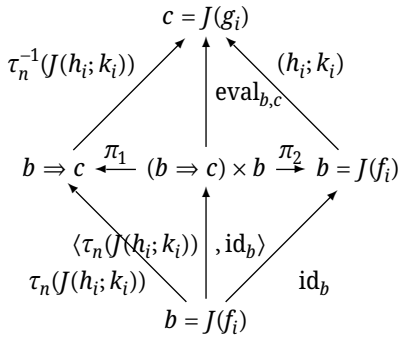
It is now enough only to verify the constraints (4.4) and (4.6), i. e., for each arrow $(h_i; k_i) : J(f_i) \to J(g_i)$ of $\mathbf{ND}_n$, there exists the arrow $k : J(g_i) \to (J(f_i) \Rightarrow J(g_i))$ and $\mathrm{eval}_{J(f_i), J(g_i)}$:

$(J(f_i) \Rightarrow J(g_i)) \times J(f_i) \to J(g_i)$. In fact in $\mathbf{ND}_n$ the diagram (4.20) becomes, for $b = J(f_i)$, $c = J(g_i)$, and hence, from

$$\begin{aligned} J(m_1) \times J(m_2) &\triangleq J(m_1 \times m_2) \\ (b \Rightarrow c) = \big(J(f_i) &\Rightarrow J(g_i)\big) \\ &= Te_n^0(J(h_i; k_i)) \\ &= J\big(Te_{n-1}^1(f_i; g_i)\big) \\ &= J(f_i \Rightarrow g_i), \end{aligned} \tag{4.38}$$

and

$$\begin{aligned} (b \Rightarrow c) \times b &= J(f_i \Rightarrow g_i) \times J(f_i) \\ &= J\big(fi \times (f_i \Rightarrow g_i)\big), \end{aligned}$$

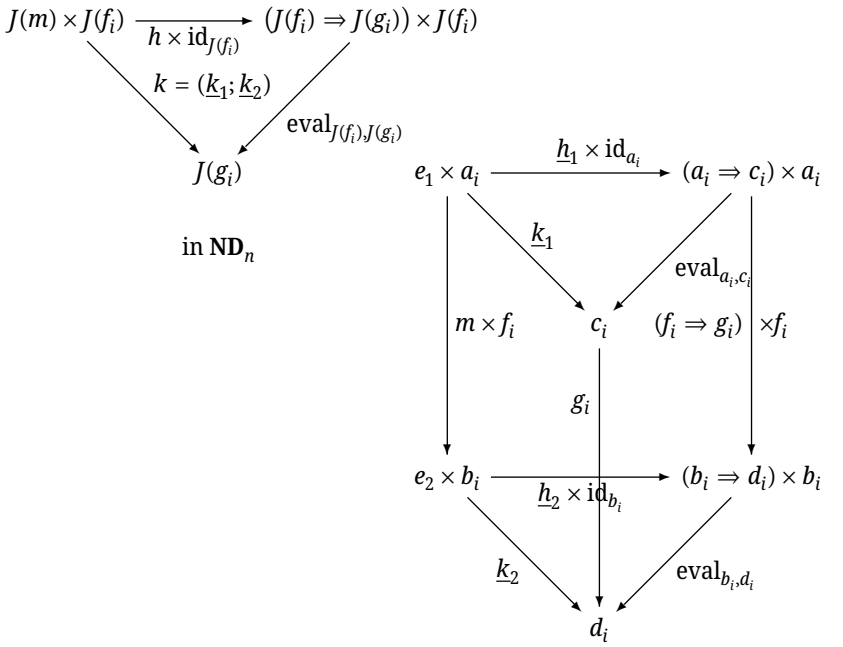


so that the "evaluation" arrow $\mathrm{eval}_{b,c} = \tau_n^{-1}(J(h_i; k_i)) \circ \pi_1 = (h_i; k_i) \circ \pi_2$ there exist as specified in (4.6), and inverse of the isomorphism $k = (\tau_n^{-1}(J(h_i; k_i)))^{-1} : J(g_i) \to (J(f_i) \Rightarrow J(g_i))$ as required by (4.4).

Let us assume the existence of the exponential diagrams in $\mathbf{ND}_{n-1}$ and show that they there exist also in $\mathbf{ND}_n$, for the given objects $J(f_i) = \langle a_i, b_i, f_i \rangle$, $J(g_i) = \langle c_i, d_i, g_i \rangle$ provided in diagram 4.35, and object $J(m) = \langle e_1, e_2, m \rangle$, and consider their exponential diagram in $\mathbf{ND}_n$ below, with the composed object

$$\begin{aligned} J(m) \times J(f_i) &= J(m \times f_i) \quad \text{from (4.38)} \\ \big(J(f_i) \Rightarrow J(g_i)\big) \times J(f_i) &= J(f_i \Rightarrow g_i) \times J(f_i) \\ &= J\big((f_i \Rightarrow g_i) \times f_i\big) \end{aligned}$$

where $(f_i \Rightarrow g_i) \triangleq Te_{n-1}^1(f_i; g_i)$ and the arrows $k = (\underline{k}_1; \underline{k}_2)$ and $h = (\underline{h}_1; \underline{h}_2)$,



(4.39)

with its equivalent commutative diagram in $\mathbf{ND}_{n-1}$ in the right-hand part, composed by two exponential (top and bottom triangles) diagrams, where from the inductive hypothesis we have that $\underline{h}_1 = \Lambda(\underline{k}_1)$ and $\underline{h}_2 = \Lambda(\underline{k}_2)$, and hence, also in the higher level $\mathbf{ND}_n$ we obtain that $h = (\underline{h}_1; \underline{h}_2) = (\Lambda(\underline{k}_1); \Lambda(\underline{k}_2)) = \Lambda(\underline{k}_1; \underline{k}_1)$, i. e., $h = \Lambda(k)$ in $\mathbf{ND}_n$ as well.

Moreover, we have that $\mathrm{eval}_{J(f_i),J(g_i)} = (\mathrm{eval}_{a_i,c_i}; \mathrm{eval}_{b_i,d_i})$, and hence, it holds that the left-hand triangle diagram in $\mathbf{ND}_n$ is an exponential diagram as well. From the fact that the exponential diagrams in the higher n-dimensional levels are composed by two exponential diagrams of $(n-1)$-dimensional level, for all n-dimensional levels we have the same partiality as for the exponential diagrams in the basic natural deduction category **ND** (with p-universal arrows and p-adjunctions).

That is, all n-dimensional levels $\mathbf{ND}_n$, for $n \geq 2$ are the pCCC. □

This concept of n-dimensional level stratification is particularly interesting for a consideration of metatheories. For example, in the basic natural deduction category **ND** the objects are positive propositional formulae while the arrows are valid deductions. So, w. r. t. the theory of propositional calculus, the objects (logical propositions) belong to the language of this theory, while the logical derivations (the arrows of **ND**) belong to the metatheory. Consequently, in the higher level $\mathbf{ND}_2 = \mathbf{ND} \downarrow \mathbf{ND}$ all objects are just the natural deduction, while the arrows between them represent the commutative deductive diagrams in **ND**.

This theorem explains that also the metatheories in effect preserve all valid deductions expressed in basic natural deduction category **ND** , only in more compressed form, because an arrow in the n-dimensional level $\mathbf{ND}_n$, for $n \geq 2$, generally expresses a number of correlated commutative deductive diagrams.

4.3 Symmetry-extended category IC for propositional intuitionistic calculus

The axioms and inference rules of the system Σ_M (*minimal positive propositional logic*) are used in order to obtain the natural deduction category **ND** as a Cartesian pCCC, so that we did not use the intuitionistic absurdity rule (where the constant $\perp$ is introduced for absurdity or *falsehood*) in (4.1),

$$\frac{\perp}{a} \quad (\perp \rightarrow)$$

There are two reasons for it:

– The first is that each natural deduction category is defined by an initial set of assumptions Γ, considered as true propositional formulae (the objects in **ND**), and all other true formulae (the objects) that are derived in constructive way. So, $\perp$ as the falsehood cannot be an object in **ND**.
– The second is that the rule above is not strictly a constructive deduction, because for a constructive deduction of b from a, we use the deduction of a to derive the formula b, and *a is not* derivable in logically consistent systems. Because of that, we do not define the negation $\neg a$ by the logical implication $a \Rightarrow \perp$, because it would be an (true) object in **ND** only if there exists an arrow (natural deduction) $g : a \rightarrow \perp$ in **ND**, that is impossible.

Consequently, from the reasons explained above, we cannot use the object (falsehood) $\perp$ to define the logical negation operation, and we remained only on the minimal positive propositional logic with inference system Σ_M.

Thus, in order to introduce the negation logical operation $\neg$ as well, we have to pass to the Gentzen sequent system, with two new inference rules for the negation operation. In what follows, we will use Gentzen LJ sequent system of logic inference rules.

Sequent calculi can be motivated by the desire to obtain more “symmetric” systems, but also systems in which proof search is easier to perform (due to the subformula property). At first, the cut rule is totally unnecessary and even undesirable, since we are trying to design systems as deterministic as possible. It has been shown that for every proof in intuitionistic logic the sequent calculus can be converted into a natural deduction proof as well. In order to provide a transformation in the other direction, we introduce the cut rule. The mismatch is resolved by Gentzen’s fundamental cut elimination theorem, which in turn singles out the crucial role played by the contraction rule. Indeed,

the contraction rule plays a crucial role in the proof of the cut elimination theorem, and furthermore, it cannot be dispensed within traditional systems for intuitionistic logic (however, in the case of intuitionistic *propositional* logic, it is possible to design contraction-free systems). So, we consider "contraction-free" systems for intuitionistic propositional logic for which the proof search always terminates.

Thus, from the categorial point of view, it means that the natural deduction category **ND**, developed for the minimal intuitionistic logic Σ_M without negation, has to be conservatively extended into the (full) propositional intuitionistic category **IC**, by introduction two new constraints for the arrows, in order to satisfy the two new inference rules for introduction and elimination of negation logical operation $\neg$ of the LJ sequent system.

In Section A.1.1 in the Appendix, there is an example of introduction of the simpler form of binary sequent calculus, where each sequent is of the form $a \vdash b$ for two given propositional formulae a and b, while the Gentzen sequent system uses the set of propositions on the left-hand side of the sequent symbol $\vdash$, i. e., sequent of the form $\Gamma \vdash b$ where $\Gamma = \{a_1, \ldots, a_n\}$ is the set of assumptions. However, we just use the symmetric tensor product $\otimes : \mathbf{ND} \times \mathbf{ND} \to \mathbf{ND}$ for the interpretation of the structural "comma" syntax operation [11] in the natural deduction category **ND**.

Thus, we are ready to translate the inference rules of the LJ sequent system into categorial structure of **ND**. This upgrade of **ND**, extended by the logical negation operation as well, will be denoted as **IC** category, with this tensor product $\otimes$, so that a *provable* sequent $a_1, \ldots, a_n \vdash b$ is then associated with an arrow (in accordance with this categorial interpretation specified by Definition 16 in Section 4.1.2),

$$h : \mathbf{c} \to b$$

where a_i and b the objects corresponding to the propositional formulae (as in category **ND**) of this sequent and $\mathbf{c} = a_1 \otimes \cdots \otimes a_n$ is a tensor product (which is not a logical formula). Since the comma is associative and commutative structural operation of a sequent, $\otimes$ is a symmetric tensor product such that, in **ND**, for any three objects a, b and c (also nonlogical formulae), we have the following arrows:

$$\mathrm{is}_a : a \simeq a \otimes a, \quad \mathrm{exch}_{a,b} : a \otimes b \to b \otimes a \quad \text{and} \quad \mathrm{assoc}_{a,b,c} : a \otimes (b \otimes c) \to (a \otimes b) \otimes c. \tag{4.40}$$

Thus, the structural inference rules for the associativity and exchange,

$$\frac{a, (b, c) \vdash d}{(a, b), c \vdash d} \quad \text{(associativity)}, \qquad \frac{\Gamma, a, b \vdash d}{\Gamma, b, a \vdash d} \quad \text{(exchange)}$$

can be represented by the **ND** category rules for the arrows:

$$\frac{f : a \otimes (b \otimes c) \to d}{f \circ \mathrm{assoc}_{a,b,c} : (a \otimes b) \otimes c \to d} \quad \text{(associativity)},$$

$$\frac{f : \mathbf{c} \otimes (a \otimes b) \to d}{f \circ (\mathrm{id}_{\mathbf{c}} \otimes \mathrm{exch}_{a,b}) : \mathbf{c} \otimes (b \otimes a) \to d} \quad \text{(exchange)}$$

while the *left weakening* structural rule is implemented in **ND** by

$$\frac{\Gamma \vdash b}{\Gamma, a \vdash b} \quad \text{(LW)} \quad \mapsto \quad \frac{f : \mathbf{c} \to b}{f \circ (\pi_1 \circ \pi) : \mathbf{c} \otimes a \to b}$$

However, different from actual *nonconstructive* well-defined interpretation,[2] the interpretation of the logical implication $\Rightarrow$ is not more than the functor right adjoint to the functor $\otimes$, but to the well-defined closure functor $T_e = (T_e^0, T_e^1) : \mathbf{ND} \downarrow \mathbf{ND} \to \mathbf{ND}$.

In the sequent system below, we will have both the cut and contraction inference rules; also, it is known that for the propositional intuitionistic logic, they are not necessary. We leave this system extendable to the logic quantifiers as well, and will show that all structural rules can be supported by our **ND** category. However, the inference rules for the negation logical connective $\neg$, not present in natural deduction, will necessarily introduce the new constraints for **ND** category, so we need to extend it into the more powerful category **IC**, in what follows. Having this in mind, let us consider the LJ sequent system for *propositional* intuitionistic logic:

1. The axiom:

$$a \vdash a \quad \text{(R)}$$

2. The structural inference rules:

$$\frac{\Gamma \vdash a, \quad \Gamma, a \vdash b}{\Gamma \vdash b} \quad \text{(cut)}, \qquad \frac{\Gamma, a, a \vdash b}{\Gamma, a \vdash b} \quad \text{(contraction)}$$

3. The inference rules for logical connectives (introduction and elimination):

$$\frac{\Gamma, a \vdash b}{\Gamma \vdash a \Rightarrow b} \quad (\to\Rightarrow), \qquad \frac{\Gamma \vdash a, \quad \Gamma, b \vdash d}{\Gamma, a \Rightarrow b \vdash d} \quad (\Rightarrow\to)$$

$$\frac{\Gamma \vdash a, \quad \Gamma \vdash b}{\Gamma \vdash a \times b} \quad (\to \times), \qquad \frac{\Gamma, a \vdash d}{\Gamma, a \times b \vdash d} \quad (\times \to)_1, \qquad \frac{\Gamma, b \vdash d}{\Gamma, a \times b \vdash d} \quad (\times \to)_2$$

$$\frac{\Gamma \vdash a}{\Gamma \vdash a + b} \quad (\to +)_1, \qquad \frac{\Gamma \vdash b}{\Gamma \vdash a + b} \quad (\to +)_2, \qquad \frac{\Gamma, a \vdash d, \quad \Gamma, b \vdash d}{\Gamma, a + b \vdash d} \quad (+ \to)$$

$$\frac{\Gamma, a \vdash}{\Gamma \vdash \neg a} \quad (\to \neg), \qquad \frac{\Gamma \vdash a}{\Gamma, \neg a \vdash} \quad (\neg \to)$$

---

2 If we define, as proposed up to now, that $\Rightarrow$ is the right adjoint for functor $\otimes : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$, i. e., by a functor $\Rightarrow = (\Rightarrow^0, \Rightarrow^1) : \mathbf{C} \to \mathbf{C} \times \mathbf{C}$, then for any object corresponding to an atomic formula $d \in \mathbf{C}$, different from a tensor product of two formulae, the unique pair (object), which introduces the implication can be $\Rightarrow^0 (d) = (d \Rightarrow d, d \Rightarrow d)$, i. e., only the "reflexive" implications $d \Rightarrow d$.
Moreover, this previous interpretation is not a constructive interpretation of the implication, as it has been satisfied for the intuitionistic logic.

With this categorial interpretation of the sequents in the **ND** category, we obtain the following transformation of the sequent system above (except for the negation logical operation ¬) in the categorial language:

1. Axiom (R) corresponds to the identity arrows $\mathrm{id}_a : a \to a$ of the category.
2. The inference rule for the cut and contraction inference rule with the isomorphisms $\mathrm{is}_a : a \to a \otimes a$ in (4.40) is represented by

$$\frac{k : \mathbf{c} \to a, \quad h : \mathbf{c} \otimes a \to b}{h \circ (\mathrm{id}_{\mathbf{c}} \otimes k) \circ \mathrm{is}_{\mathbf{c}} : \mathbf{c} \to b} \quad \text{(cut)}, \qquad \frac{k : \mathbf{c} \otimes a \otimes a \to b}{k \circ (\mathrm{id}_{\mathbf{c}} \otimes \mathrm{is}_a^{-1}) : \mathbf{c} \otimes a \to b} \quad \text{(contraction)}$$

3. The inference rule for introduction of implication (→⇒), for the precondition arrow $f : a \to b$ of **ND**, which defines the implication object $(a \Rightarrow b) = T_e^0(J(f))$ and the arrow $k = f \circ \pi_2 : \mathbf{c} \times a \to b$, is represented by

$$\frac{k \circ \pi : \mathbf{c} \otimes a \to b}{\Lambda(k) : \mathbf{c} \to a \Rightarrow b} \quad (\to\Rightarrow)$$

   where from (4.2) we have the arrow $\pi : \mathbf{c} \otimes a \to \mathbf{c} \times a$, and from (4.12) the "abstraction" $\Lambda(k)$ of the arrow k.
4. The inference rule for elimination of implication (⇒→) is represented by

$$\frac{h : \mathbf{c} \to a \quad k : \mathbf{c} \otimes b \to d}{k \circ g : \mathbf{c} \otimes (a \Rightarrow b) \to d} \quad (\Rightarrow\to)$$

   where $g = (\mathrm{id}_{\mathbf{c}} \otimes h_1) \circ (\mathrm{is}_{\mathbf{c}} \otimes \mathrm{id}_{a \Rightarrow b})$ with the arrows $\mathrm{is}_{\mathbf{c}} : \mathbf{c} \to \mathbf{c} \otimes \mathbf{c}$ and

$$h_1 = \mathrm{eval}_{a,b} \circ \pi \circ (\mathrm{id}_{a \Rightarrow b} \otimes h) \circ \mathrm{exch}_{\mathbf{c}, a \Rightarrow b} : \mathbf{c} \otimes (a \Rightarrow b) \to b,$$

   where from (4.2), we have the nondeductive arrow $\pi : (a \Rightarrow b) \otimes a \to (a \Rightarrow b) \times a$. Note that, from the extended symmetry property of **ND**, also $g = \mathrm{id}_{\mathbf{c}} \otimes \tau^{-1}(J(f))$ for $\tau^{-1}(J(f)) : (a \Rightarrow b) \to b$, where $f : a \to b$ is an arrow for which there exists implication $(a \Rightarrow b) = T_e^0(J(f))$ in **ND**.
5. The inference rule for introduction of product (→ ×) is represented by

$$\frac{k_1 : \mathbf{c} \to a \quad k_2 : \mathbf{c} \to b}{\langle k_1, k_2 \rangle : \mathbf{c} \to a \times b} \quad (\to \times)$$

   as it follows from the commutative diagram for the product (limit) cone (4.13).
6. The two inference rules for elimination of product (× →) are represented by

$$\frac{k : \mathbf{c} \otimes a \to d}{k \circ (\mathrm{id}_{\mathbf{c}} \otimes \pi_1) : \mathbf{c} \otimes (a \times b) \to d}, \qquad \frac{k : \mathbf{c} \otimes b \to d}{k \circ (\mathrm{id}_{\mathbf{c}} \otimes \pi_2) : \mathbf{c} \otimes (a \times b) \to d}$$

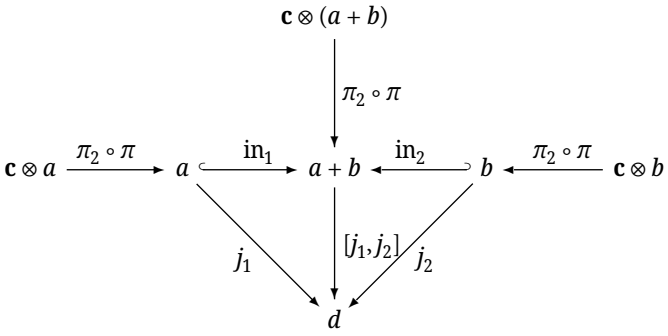
7. The inference rules for introduction of coproduct (→ +) is represented by (see the commutative diagram of coproduct cocone (4.10))

$$\frac{k_1 : \mathbf{c} \to a}{\mathrm{in}_1 \circ k_1 : \mathbf{c} \to a + b} \quad (\to +)_1, \qquad \frac{k_2 : \mathbf{c} \to b}{\mathrm{in}_2 \circ k_2 : \mathbf{c} \to a + b} \quad (\to +)_2$$

8. The inference rule for elimination of coproduct $(+ \to)$ is a version of that used for the sequent-based natural deduction with two precondition arrows j_1 and j_2, and hence, represented by

$$\frac{j_1 \circ \pi_2 \circ \pi : \mathbf{c} \otimes a \to d \quad j_1 \circ \pi_2 \circ \pi : \mathbf{c} \otimes b \to d}{[j_1, j_2] \circ \pi_2 \circ \pi : \mathbf{c} \otimes (a + b) \to d} \quad (+ \to)$$

as follows from the extension of the cocone coproduct diagram (4.10), analog to that in scheme (4.14), of the coproduct $a + b$ in **ND**, such that for any object d with cocone component conditional arrows $j_1 : a \to d$ and $j_2 : b \to d$, there is a unique arrow $[j_1; j_2] : a + b \to d$, as follows:



Thus, as we can see, the positive part of the intuitionistic logic (without negation operation) is an extension of **ND** with structural deductive arrows in (4.40). The arrows in this extension of the **ND** model its inference rules for all logical connectives different from negation ¬. That is, **ND** is not only a categorial model for the natural deduction for positive propositional logic but a model for the positive intuitionistic logic as well. So, the traduction of the sequent-based inference rules is of this type:

"*If there is a particular arrow in* ***ND****, then must exist another particular arrow.*"

However, such a kind of categorial inference is not adequate for the inference rules for introduction and elimination of logical negation ¬, from the fact that in them we have the sequents with an *empty* right-hand side, which cannot be represented by an arrow in **ND**, because each arrow must have codomain object as well. Thus, we have the following.

Definition 21. The inference rules for the introduction and elimination of logical negation operations has to be traduced into categorial language as the following two *new* constraints are not supported by **ND** (we denote by **IC** this extension w. r. t. category **ND**):

1. The inference rule $(\to \neg)$ is traduced by the constraint, for a given object **c** of **IC** if a formula a is not an object of **IC**, and if from **c** and a cannot be inferred any formula, then we can insert the formula $\neg a$ as a new object of **IC**, with the arrow

$$\mathrm{neg}_a : \mathbf{c} \to \neg a. \tag{4.41}$$

2. The inference rule $(\neg \to)$ is traduced by the constraint that, for a given arrow $f : \mathbf{c} \to a$ of **IC** where the object a is a logic formula, the logic formula $\neg a$ cannot be inserted into this category **IC**.

So, we obtain the final result.

Corollary 11. *he category* **IC**, *obtained as an extension of the* **ND** *with negation operation with two constraints of Definition* 21*, and with structural deductive arrows in* (4.40)*, is a categorial model of Intuitionistic propositional logic (IL).*

Proof. The proof is directly from the categorial translation of the intuitionistic inference rules above and Definition 21. □

5 Internal symmetry and lambda calculus

5.1 Introduction to lambda calculus

As we have seen in Chapter 4 in the case of the propositional intuitionistic logic and its proof theory, the leading theme has been the so-called Curry–Howard correspondence, known as the "formulae-as-types" analogy as well, which exploits the connection between proof theory and type theory, and has been the main methodological tool both for understanding the relation between intuitionistic logic and typed lambda calculus, and for the design of new functional languages with powerful-type systems.

The lambda calculus notation has been explicitly introduced by Alonzo Church [20], a formalism provided to develop a general theory of computable functions (Church defined the notion of computable function via this system). At first, its expressions represent both a program and its data, which on their turn, evolve by simple rules (β and η-reduction) to instantaneous descriptions of intermediate computational results possibly ending in output. These features, however, are also present in term rewriting systems. Lambda calculus has two extra qualities setting it apart from these other rewriting systems:

– First, it is applicative, in that an expression may be applied to another expression, giving them the possibility to act both as a function and as an argument.
– Second, lambda calculus has abstraction built in, meaning that the expressions are closed under explicit definitions.

Lambda calculus as a model of computation can be introduced in an untyped fashion: arbitrary expressions may be applied to each other. A similarly flat approach to computing was present in the early assembly languages. In computer science, the lambda calculus made its first appearance in programming language LISP, which has been also the first formal programming language to use the procedures as objects of the language. Consequently, in LISP the programs and data are not distinguished, but they are all elements of a unique untyped universe (the universe of λ-terms for the λ-calculus, called the universe of S-expressions in LISP). Anyway, in each programming practice, the types arise naturally, even starting from untyped universes, when the programming objects are categorized according to their usage or behavior. Since the first appearance of types in the programming language Algol 60, when the typing of variables was introduced to check at compile time the connection of instances of use with associated declarations, typing has been considered an essential method that guides a programmer in a design of a programming code.

Functional programming languages, like Miranda, ML, etc. are based on lambda calculus. An early (although somewhat hybrid) example of such a language is LISP as explained above. *Reduction machines* are specifically designed for the execution of these

https://doi.org/10.1515/9783111081670-005

functional languages in an analog way as the Turing machines[1] (1936), which defined the notion of a computable function via these machines.

The first basic operation of the λ-calculus is *application*, with expression $M \cdot N$ or MN, which denotes the data M as an algorithm applied to the data N. The theory is type-free: it allows us to consider expressions like MM, i. e., M is applied to itself. The other basic operation is *abstraction*: if M is an expression containing a variable x (e. g., $5*x-3*y+32$), then $\lambda x.M$ denotes the function $x \mapsto M$. Abstraction is said to *bind* the free variable x in M (the variable y in M remains free). Substitution is only performed for the occurrences of free variables as usual. Lambda calculus, as treated so far, is usually referred to as a type-free theory. This is so, because every expression (considered as a function) may be applied to every other expression (considered as an argument). For example, the identity function $I \equiv \lambda x.x$ may be applied to any argument x to give as a result that same x. In particular, I may be applied to itself.

There are also typed versions of lambda calculus. These are introduced essentially in Curry [21] (for the so-called combinatory logic, a variant of lambda calculus) and in Church [22]. Types are usually objects of a syntactic nature and may be assigned to lambda terms. If M is such a term and a type a is assigned to M, then we say "M has type a" or "M in a"; the denotation used for this is $M : a$.

Although the analogy is not perfect, the type assigned to a term may be compared to the dimension of a physical entity. These dimensions prevent us from wrong operations like adding 3 volts to 2 ampere. In a similar way, types assigned to lambda terms provide a partial specification of the algorithms that are represented and are useful for showing partial correctness. Types may also be used to improve the efficiency of compilation of terms representing functional algorithms. If, for example, it is known (by looking at types) that a subexpression of a term (representing a functional program) is purely arithmetical, then fast evaluation is possible.

The Curry and Church approaches to typed lambda calculus correspond to two paradigms in programming. In the first of these, a program may be written without typing at all. Then a compiler should check whether a type can be assigned to the program. This will be the case if the program is correct. A well-known example of such a language is ML. The style of typing is called *implicit typing*.

The other paradigm in programming is called *explicit typing* and corresponds to the Church version of typed lambda calculi. Here, a program should be written together with its type. For these languages, type-checking is usually easier, since no types have to be constructed. Examples of such languages are Algol 68 and Pascal. It is the case that terms that do have a type always possess a normal form. By the unsolvability of the

1 Based on the concept of a Turing machine are the present day von Neumann computers. Conceptually, these are Turing machines with random access registers. Imperative programming languages such as Fortran, Pascal, etc. as well as all of the assembler languages are based on the way a Turing machine is instructed: by a sequence of statements.

halting problem this implies that not all computable functions can be represented by a typed term. This is not so bad as it sounds, because in order to find such computable functions that cannot be represented, one has to stand on one's head.

Considering terms and types as programs and their specifications is not the only possibility. A type a can also be viewed as a proposition and a term M in a as a proof of this proposition. This so-called *propositions-as-types* interpretation is independently due to de Bruijn [23] and Howard [24] (both papers were conceived in 1968). Hints in this direction were given in Curry and Feys [25].

The Church-typed lambda calculus is a collection of type labels TL, over a ground set of atomic type symbols At, and hence is defined inductively by

– $At \subseteq TL$;
– If $a, b \in TL$, then $a \Rightarrow b \in TL$;
– If $a, b \in TL$, then $a \times b \in TL$.

For every type $a \in TL$, there exists a denumerable number of variables, ranged over by lowercase letters near the end of alphabet, $x, y, z, \ldots$, while the metavariables for terms are used upper case letters $M, N, P, \ldots$. So, the fact that a term M has type a is denoted with expression $M : a$. The well-defined terms (w. d. t) and their associated types, are defined according to the following formation rules:

1. every variable $x : a$ is a w. d. t.
2. if $x : a$ is a variable, and $M : b$ is a w. d. t., then $\lambda x : a.M : a \Rightarrow b$ is a w. d. t.
3. if $M : a \Rightarrow b$ is a w. d. t and $N : a$ is a w. d. t., then $MN : b$ is a w. d. t.
4. if $M : a$ is a w. d. t and $N : b$ is a w. d. t., then $\langle M, N\rangle : a \times b$ is a w. d. t.
5. if $M : a \times b$ is a w. d. t, then $\mathrm{fst}(M) : a$ is a w. d. t.
6. if $M : a \times b$ is a w. d. t, then $\mathrm{snd}(M) : b$ is a w. d. t.

For more details about the free-variables of terms and the substitution $[M/x]N : b$ of a proof $M : a$ for a generic $x : a$ in a proof $N : b$, a reader can use a more specific books dedicated to typed lambda calculus.

We consider an equational theory of proofs, defined as the minimal congruence relation $'='$ which satisfies the following axiom schemas:

(α) $\lambda x : a.M = \lambda y : a.[y/x]M$
($\rightarrow \beta$) $(\lambda x : a.M)N = [N/x]M$
($\rightarrow \eta$) $\lambda x : a.(Mx) = M$, if x is not a free variable
($\times\beta_1$) $\mathrm{fst}(\langle M, N\rangle) = M$
($\times\beta_2$) $\mathrm{snd}(\langle M, N\rangle) = N$
($\times\eta$) $\langle \mathrm{fst}(P), \mathrm{snd}(P)\rangle = P$

If we read these equations from left to right, then we obtain a rewriting system, which defines the operational semantics of typed lambda calculus as a programming language.

The rewriting rules associated with ($\rightarrow \beta$), ($\times\beta_1$) and ($\times\beta_2$) are called β-reductions, and those associated with ($\rightarrow \eta$) and ($\times\eta$) are called η-reductions. From the point of

view of the Curry–Howard correspondence, both η-reductions and β-reductions (which play a major rule) have an interpretation in terms of proof normalization. A term is in *normal form* (resp., β-normal form), if it is no more reducible (resp., β-reducible): the β-reduction relation is confluent and Noetherian.

So, it is easy to see that the categorial model for the typed lambda calculus is given by a category with objects equal to lambda types and arrow to lambda terms, such that $\times$ is the categorial product and for each to objects (lambda types) we have the exponential object $a \Rightarrow b$, while fst and snd are represented by the first and second product projections π_1 and π_2, respectively. The following result is well known [11, 26]:

> *"Any CCC with enough points* $\mathbf{C}$ *and any collection* $a_i \in At$ *of objects in* $\mathbf{C}$ *yield a model of typed lambda calculus* $\lambda\beta\eta\pi^{\mathbf{t}}$*."*

The typed lambda calculus is *strongly normalizable*, i. e., every computation terminates and the computation is independent from the evaluation strategy: thus, in the design of programming code one does not have to consider the operational evolution of reduction, but to concentrate only to the denotation of the program. Obviously, not all Turing-computable functions (e. g., the Ackermann functions) are computable by typed lambda calculus language, but it does not worry anybody: the primitive recursive functions are enough for most interesting applications.

Example 14 (Sequent system for intuitionistic logic). Let us consider the sequent systems of natural deduction and propositional intuitionistic logic considered in Chapter 4.

There is a bijective correspondence between logical formulae and types of the typed lambda calculus with explicit pairs. We can associate every formula b with a λ-term M of the respective type, which intuitively represent its proofs. In particular, if $x_1 : a, \ldots, x_n : a_n$ are the free variables in M, then $M : b$ is a proof of M depending on the hypothesis $\Gamma = \{a_1, \ldots, a_n\}$. Thus, if M is a proof of b, possibly depending on a generic proof $x : a$, one gets a proof of $a \Rightarrow b$ by "abstraction on $x : a$," i. e., $\lambda x : a.M : a \Rightarrow b$. Juxtaposition of proofs corresponds to their sequential composition. So, if we have a proof $M : a \Rightarrow b$, and we apply it to a proof $N : a$, then we obtain a proof MN of b. A proof of $a \times b$ is given by a pair of distinct proofs for a and b.

The inference rules of calculus exactly formalize this process of constructing the complex proofs by means of simpler ones. A variable $x : a$ is a generic proof of a. So, an intuitionistic sequent in Section 4.3 has the following λ-calculus syntactic structure:

$$x_1 : a_1, \ldots, x_n : a_n \vdash M : b$$

where $x_1 : a_1, \ldots, x_n : a_n$ is a finite list of distinct generic proofs, and $M : b$ is a proof of b whose free variables are among $x_1, \ldots, x_n$. Every formula in the left-hand side has an associated distinct variable, and hence no confusion can arise between formulae, even if they have the same name (symbol).

The intuitive interpretation of the sequent $\Gamma \vdash M : b$, for $\Gamma = x_1 : a_1, \ldots, x_n : a_n$, is that of a process, which builds a proof M of b, as soon as it has the proofs for $a_1, \ldots, a_n$.

If $\Gamma \vdash M : b$ is the end sequent of a cut-free derivation (deduction), then the proof $M : b$ is in β-normal form: The cut-elimination theorem shows that if D is a derivation of $\Gamma \vdash M : b$. Then there is a cut-free derivation D' of the sequent $\Gamma \vdash N : b$, where N is the β-normal form of M.

It is possible to show by pure simply typed λ-calculus can compute all the arithmetic functions, i. e., all the functions that are defined by compositions of the previous ones (the predecessor function is not representable, for example). However, simply typed λ-calculus cannot be considered as a real programming language, but only as a paradigm for strongly typed functional programming languages. There are two types of extensions:

– The first is a "polimorphic" extension (of the "second order"), which still has the nice property of strong normalization together with a great formal expressiveness, but imposes to the programmer a completely new approach to design of the programming code.
– The second is a simple extension by means of fixpoints operators, which enables us to write recursive definitions of functions.

Models of the untyped λ-calculus may be defined either as *reflexive objects* in Cartesian closed categories (categorial models) or as combinatory algebras satisfying the five axioms of Curry and the Meyer–Scott axiom (λ-models). Concerning categorial models (interesting in our approach), it has been shown that each of them can be presented as a λ-model, even when the underlying category does not have enough points.

Next, we will consider an extension of the language by *fixed-point operators*.

Definition 22. A fixed point of type b is a term Θ_b of type $(b \Rightarrow b) \Rightarrow b$ such that, for every term $M : b \Rightarrow b$, we obtain

$$M : b \Rightarrow b(\Theta_b M : b \Rightarrow b) = \Theta_b M : b \Rightarrow b. \tag{5.1}$$

By fixed-point operators, we can simulate recursive definitions. The combined use of these operators with a reasonable set of arithmetic primitives gives to this language its *full computational power*. However, such an extension has a consequence that the fixed-point operators have no logical formula correspondent which limits the Curry–Howard analogy. That is, by introducing the fixed-point operators we obtain the full expressive power and the elegance of programming by "recursive definitions" but cannot use more the "types as formulae" paradigm.

5.2 Reflexive objects in the Cartesian closed categories

It has been shown by Lambek that Cartesian Closed Categories (CCC) can be identified with extensional typed λ-calculus and another interesting connection between CCC and λ-calculus is Scott's embedding of λ-theory into a CCC (referred as Scott embedding) [27, 28]. So, the general concept of λ-models is made clear by presenting an elegant first-order presentation. This is done by using yet another equivalent notion, given by several authors; namely, the functional λ-models which is a set b and collection of functions $b \Rightarrow b$ (which are formally definable) on it such that $b \Rightarrow b < b$ (so that b is a reflexive object with enough points in a CCC, and hence a strong motivation for the categorial approach).

Given a λ-model, it is always possible to define a CCC where it can be viewed as a reflexive object with enough points [11], and by using Koymans construction [29] based on work of Scott, arbitrary reflexive object in a CCC give rise to λ-algebras and to all of them: it turns out that the λ-models are *exactly* those λ-algebras that come from reflexive objects with enough points. It was demonstrated that *any* reflexive object of an arbitrary CCC gives rise to a λ-model, when choosing in Koymans construction appropriately the underlying combinatory algebra (e. g., in [30, 31] the authors provide reflexive objects in categories of games, hence also without enough points).

Consequently, in order to define a categorial interpretation of the fixed-point operators, we consider the CCC which are categorial models of the lambda calculus, and define new constraints that they has to satisfy in order to be able to represent the fixed-point operators and the equations (5.1), which in a category are expressed by the commutative diagrams where the lambda terms are represented by the arrows and their types by the objects. Hence, we introduce the following definition.

Proposition 10. *Let* **C** *be a CCC. A fixed-point operator, for a given object b of* **C**, *is an arrow* $\mathrm{fix}_b : (b \Rightarrow b) \to b$ *such that* $\mathrm{fix}_b = \mathrm{eval}_{b,b} \circ \langle \mathrm{id}_{b\Rightarrow b}, \mathrm{fix}_b \rangle : (b \Rightarrow b) \to b$, *corresponding to the following commutative diagram:*

$$\begin{array}{ccc} (b \Rightarrow b) \times b & \xrightarrow{\mathrm{eval}_{b,b}} & b \\ \uparrow{\scriptstyle \langle \mathrm{id}_{b\Rightarrow b}, \mathrm{fix}_b \rangle} & \nearrow{\scriptstyle \mathrm{fix}_b} & \\ b \Rightarrow b & & \end{array} \tag{5.2}$$

such that

$$\mathrm{fix}_b = \mathrm{eval}_{b,b} \circ \langle \mathrm{id}_{b\Rightarrow b}, \mathrm{in}_2 \rangle \circ \Lambda(\mathrm{eval}_{b,b} \circ (\mathrm{id}_{b\Rightarrow b} \times (\mathrm{eval}_{b,b} \circ \langle \mathrm{id}_{b\Rightarrow b}, \mathrm{in}_2 \rangle \circ \mathrm{out}_2))). \tag{5.3}$$

A category has fixed-point operators if each object b has a fixed-point operator fix_b.

Proof. Let us denote $\mathrm{fix}_b = h_1 \circ h_2$ as composition of two morphisms, $h_1 = \mathrm{eval}_{b,b} \circ \langle \mathrm{id}_{b\Rightarrow b}, \mathrm{in}_2\rangle : (b \Rightarrow b) \to b$ and $h_2 = \Lambda(\mathrm{eval}_{b,b} \circ (\mathrm{id}_{b\Rightarrow b} \times (h_1 \circ \mathrm{out}_2)))$, so that we have

$$\begin{aligned}
\mathrm{fix}_b &= h_1 \circ h_2 \\
&= \mathrm{eval}_{b,b} \circ \langle \mathrm{id}, \mathrm{in}_2\rangle \circ \Lambda(\mathrm{eval}_{b,b} \circ (\mathrm{id} \times (h_1 \circ \mathrm{out}_2))) \\
&= \mathrm{eval}_{b,b} \circ \langle \Lambda(\mathrm{eval}_{b,b} \circ (\mathrm{id} \times (h_1 \circ \mathrm{out}_2))), \mathrm{in}_2 \circ \Lambda(\mathrm{eval}_{b,b} \circ (\mathrm{id} \times (h_1 \circ \mathrm{out}_2)))\rangle \\
&= \mathrm{eval}_{b,b} \circ (\Lambda(\mathrm{eval}_{b,b} \circ (\mathrm{id} \times (h_1 \circ \mathrm{out}_2))) \times \mathrm{id}) \\
&\quad \circ \langle \mathrm{id}, \mathrm{in}_2 \circ \Lambda(\mathrm{eval}_{b,b} \circ (\mathrm{id} \times (h_1 \circ \mathrm{out}_2)))\rangle \\
&= \mathrm{eval}_{b,b} \circ (\mathrm{id} \times (h_1 \circ \mathrm{out}_2)) \circ \langle \mathrm{id}, \mathrm{in}_2 \circ \Lambda(\mathrm{eval}_{b,b} \circ (\mathrm{id} \times (h_1 \circ \mathrm{out}_2)))\rangle \\
&= \mathrm{eval}_{b,b} \circ \langle \mathrm{id}, (h_1 \circ \mathrm{out}_2) \circ \mathrm{in}_2 \circ \Lambda(\mathrm{eval}_{b,b} \circ (\mathrm{id} \times (h_1 \circ \mathrm{out}_2)))\rangle \\
&= \mathrm{eval}_{b,b} \circ \langle \mathrm{id}, h_1 \circ \Lambda(\mathrm{eval}_{b,b} \circ (\mathrm{id} \times (h_1 \circ \mathrm{out}_2)))\rangle \\
&= \mathrm{eval}_{b,b} \circ \langle \mathrm{id}, h_1 \circ h_2\rangle \\
&= \mathrm{eval}_{b,b} \circ \langle \mathrm{id}, \mathrm{fix}_b\rangle
\end{aligned}$$

so that diagram (5.2) commutes. □

Thus, e. g., if we consider the example of a CCC category where the object b is a set and arrows are the functions, and $b \Rightarrow b$ represent the set of all functions from the set b into the same set, for each point (a function) $g : b \to b$ of the object $b \Rightarrow b$, for the fixed-point operator (a function) fix_b, we obtain that the commutative diagram (5.2) represents, for a function g, for some $x_0 \in b$, the equation

$$\begin{aligned}
x_0 \triangleq \mathrm{fix}_b(g) &= \mathrm{eval}_{b,b} \circ \langle \mathrm{id}_{b\Rightarrow b}, \mathrm{fix}_b\rangle(g) = \mathrm{eval}_{b,b}(g, \mathrm{fix}_b(g)) = g(\mathrm{fix}_b(g)) \\
&= g(x_0),
\end{aligned}$$

and hence, $(g \circ g)(x_0) = g(g(x_0)) = g(x_0)$ so that g is idempotent. Thus, we are interesting for idempotent elements (functions) g of the exponent object $b \Rightarrow b$. In a more general setting, the existence of exponents $b \Rightarrow b$ suggests the investigation of those objects that "contain," as retract,[2] their own function space: namely, the *reflexive objects*. It has been demonstrated that if a category **C** is a CCC, then and object b is reflexive iff

$$(b \Rightarrow b) < b.$$

These nontrivial reflexive objects does not exist in each CCC; e. g., in the category **Set** which is a CCC where the object $b \Rightarrow b$ is the set of all functions from b into b, and hence is strictly bigger than the cardinality of the object (a set) b. However, if instead of (simple, unstructured) sets used as the object of a CCC, for a CCC we use the objects that have

2 A pair of arrows $f : b \to d$ and $g : d \to b$ of a category **C** is a *retraction pair* iff $g \circ f = \mathrm{id}_b$ (where it is easy to show that f is monic arrow and g is epic arrow). Then b is called *retract* of d, denoted by $b < d$ via retraction pair (f, g) with $e = f \circ g : d \to d$ a *split idempotent*.

enough reach internal structure, e. g., the set with a binary reflexive and transitive relation $\leq$ between its elements, and impose some restrictions for the well-defined arrows between such objects (e. g., to be the monotonic functions that preserves this ordering between domain and codomain object, then the cardinality of such arrows can be less than the cardinality of the arrows between these two objects considered as simple sets of elements).

For still more complex internal structures of the objects (like *Scott domains* introduced in Definition 23, or the tree structures of labeled transition systems, for example), we can impose also another restrictions for the well-defined morphisms, which preserves the internal properties of such complex object, so that the object $b \Rightarrow b$, which represent all such endomorphisms for a given object b can have significatively less cardinality from the cardinality of elements in this object b.

The retracts and exponents has been considered for definition of categorial models of type-free languages, where the programs are viewed as data or, semantically, exponents may be retracted into (source and target) objects. Note that if (g_1, g_2), for $g_1 : a \to d$ and $g_2 : d \to a$, is a retraction pair such that $g_2 \circ g_1 = \mathrm{id}_a : a \to a$, then the arrow $f = g_1 \circ g_2 : d \to d$ is a *split idempotent*: in fact,

$$\begin{aligned} f \circ f &= (g_1 \circ g_2) \circ (g_1 \circ g_2) \\ &= g_1 \circ (g_2 \circ g_1) \circ g_2 \\ &= g_1 \circ \mathrm{id}_b \circ g_2 = g_1 \circ g_2 \\ &= f. \end{aligned}$$

The most well-known example of CCC with fixed-point operators is probably the category **CPO**, with complete partial orders for objects and continuous functions for arrows.[3] We can show, e. g., that such CCC with reflexive objects can have also the categorial symmetry as, e. g., the CCC category **Set** (which, however, has no reflexive objects).

---

3 Let us introduce the Scott's domains as follows.

Definition 23. Let $(X, \leq)$ be a partially ordered set (p. o. set). $d \subseteq X$ is *directed* iff it is nonempty and for any $i, j \in d$, there is $k \in d$ such that $i \leq k$ and $j \leq k$.
A p. o. set $(X, \leq)$ is *complete* (is a **CPO**) iff every directed subset $d \subseteq X$ has a least upper bound (l.u.b) $\curlyvee d$ (the bottom element $\bot = \curlyvee\emptyset$ is the least upper bound of the empty directed set):

1. The **CPO** $(X, \leq)$ is *algebraic* if for every $j \in X$ the set $\downarrow j = \{k \in X \mid k \leq j\}$ is directed and $\curlyvee(\downarrow j) = j$.
2. A **CPO** $(X, \leq)$ is *bounded complete* if every bounded subset of X has a l. u. b.
3. A Scott domain is a bounded complete algebraic **CPO**.

In a **CPO**, a function $f : a \to b$ is *continuous* iff f is monotonic and for every directed subset d of a, $f(\curlyvee d) = \curlyvee f(d)$, and given continuous function f, we can form a chain

$$\left\{f^n(\bot)\right\}_{n\in\omega} = \bot \leq f(\bot) \leq f\big(f(\bot)\big) \leq \cdots \leq f^n(\bot) \leq \cdots$$

starting from the bottom element $\bot$.

Example 15. A CCC with conceptual categorial symmetry and with reflexive objects as well:

Let us consider a category **C** where the objects are Scott's domains, and the arrows are continuous functions between them (as specified in Definition 23). So, we can define the symmetry-closure functor $T_e : \mathbf{C} \downarrow \mathbf{C} \to \mathbf{C}$, such that for an arrow $f : a \to b$ (a continuous function between to Scott's domains), we obtain a conceptualized object $\widetilde{f} = T_e^0(J(f))$ defined by the graph of this continuous function, i. e.,

$$\widetilde{f} = T_e^0(J(f)) \triangleq \{(i, f(i)) \mid i \in \mathrm{dom}(f)\} \tag{5.4}$$

Thus, the conceptualized object are partial orders $(\widetilde{f}, \leq_f)$ with bottom element $(\bot, f(\bot))$ (where $\bot$ is bottom element of Scott's domain $a = \mathrm{dom}(f)$), determined by the partial order $(a, \leq_a)$ of the object $a = \mathrm{dom}(f)$, such that for any two elements $(i, f(i))$ and $(j, f(j))$ of the conceptualized object $\widetilde{f}$, we have that

$$(i, f(i)) \leq_f (j, f(j)) \quad \text{iff} \quad i \leq_a j, \tag{5.5}$$

and hence iff $f(i) \leq_b f(j)$, form the fact that $f : a \to b$ is a continuous, and hence monotonic, as well, function. From (5.5) and the first projection $\pi_1(\widetilde{f}) = a$, which is Scott's domain, also $(\widetilde{f}, \leq_f)$ is down-bounded and directed: for each $(i, f(i)), (j, f(j)) \in \widetilde{f}$, from the fact that $a = \mathrm{dom}(f)$ is Scott's domain (which is directed), there is $k \in a$ such that $i \leq k$ and $j \leq k$, and hence $(i, f(i)) \leq_f (k, f(k))$ and $(j, f(j)) \leq_f (k, f(k))$.

It is complete (it is a **CPO**) because for every directed subset $S \subseteq \widetilde{f}$ there exists the least upper bound $\curlyvee_f S = (\curlyvee(\pi_1(S)), f(\curlyvee(\pi_1(S))))$.

A point $(i, f(i))$ of $(\widetilde{f}, \leq_f)$ is compact (finite) if for every directed S such that $(i, f(i)) \leq_f \curlyvee S$ there is an element $(j, f(j)) \in S$ such that $(i, f(i)) \leq_f (j, f(j))$, i. e., $i \leq j$ in Scott's domain $a = \mathrm{dom}(f)$.

Let Y_0 denote the collection of compact elements of $\widetilde{f}$. The c.p.o $(\widetilde{f}, \leq_f)$ is algebraic because Scott's domain $a = \mathrm{dom}(f)$ is algebraic, i. e., for every $(i, f(i)) \in \widetilde{f}$, the set

$$\downarrow (i, f(i)) = \{(j, f(j)) \in Y_0 \mid (j, f(j)) \leq_f (i, f(i))\} = \{(j, f(j)) \mid j \leq i\}$$

is directed. In fact, for each $i \in a = \mathrm{dom}(f)$, the set $(i, f(i)) = \{(j, f(j)) \in Y_0 \mid j \in a$ is compact in a and $j \leq i\}$ is directed because $a = \mathrm{dom}(f)$ is c.p.o and

$$\curlyvee_f \downarrow (i, f(i)) = (\curlyvee(\downarrow i), f(\curlyvee(\downarrow i))) = (i, f(i))$$

because it holds that $\curlyvee(\downarrow i) = i$ from the fact that $a = \mathrm{dom}(f)$ is Scott's domain (and hence algebraic **CPO**).

So, from the fact that the poset relation $\leq_f$ preserves the poset relation $\leq_a$ of the Scott's domain $(a, \leq_a)$, and f is continuous function, it holds that also the conceptual object $\widetilde{f}$ as a partial order with $\leq_f$ satisfies all requirements as the Scott's domain $(a, \leq_a)$.

Thus, also the poset $(\widetilde{f}, \leq_f)$ is a Scott's domain (bounded complete algebraic **CPO**) with the bottom element $(\bot, f(\bot))$, and hence well-defined object of **C**.

Moreover, for each identity arrow $\mathrm{id}_a : a \to a$ (which is obviously a continuous function) of the object a (Scott's domain) of **C**, the conceptualized object obtained from it is

$$\widetilde{\mathrm{id}}_a = \{(x, \mathrm{id}_a(x)) \mid x \in a\} = \{(x, x) \mid x \in a\},$$

and hence it is isomorphic to the Scott's domain a, as required by the conceptual symmetry representation requirement, and this isomorphic arrow $\varphi(a) : \widetilde{\mathrm{id}}_a \simeq a$, which maps each element (x, x) of $\widetilde{\mathrm{id}}_a$ into the element x of a is continuous function as well in **C**.

For any commutative diagram in **C**, $g \circ k_1 = k_1 \circ f$, composed by the continuous functions, the arrow component T^1_e of the functor, for the arrow $(k_1; k_2) : J(f) \to J(g)$ defines the function $T^1_e(k_1; k_2) : \widetilde{f} \to \widetilde{g}$ in **C**, such that for any element $(x, f(x))$ of the conceptualized object $\widetilde{f}$, it holds that

$$T^1_e(k_1; k_2)(x, f(x)) \triangleq \big(k_1(x), k_2(f(x))\big) \in \widetilde{g} \tag{5.6}$$

so that, from the fact that both k_1 and k_2 are continuous functions, the function $T^1_e(k_1; k_2)$ is monotonic and continuous as well, and hence a well-defined arrow in **C**.

Note that in the same way we can show that also each **CPO**, which is a CCC, is conceptually closed category by the symmetry-closure functor T_e defined by (5.4) and (5.4). The Scott continuous semantics corresponds to the class of λ-models having cpo's (complete partial orders) as underlying sets and representing all Scott continuous functions. By applying a construction due to Koymans [29] and based on work of Scott, arbitrary reflexive objects in CCC's give rise to λ-algebras and to all of them.

Based on these considerations, we can define the categories of idempotents, by using the conceptual categorial symmetry, and hence different from the categories $\mathbf{Ret}_b$, defined previously as particular subcategories of the *Karoubi envelope* [32] obtained by fixing a single object b of a category **C**, and used for the fixed-point operators (see [11], for example). Thus, as a first step, we introduce these categories of idempotents (used by Scott in [28, 29]).

Definition 24. Given a reflexive object b in a category **C**, the category $\mathbf{Ret}_b$ of idempotents on b is so defined:

1. $\mathrm{Ob}_{\mathbf{Ret}_b} = \{f \in \mathbf{C}(b, b) \mid f \circ f = f\}$
2. $\mathrm{Mor}_{\mathbf{Ret}_b} = \{(f, k, g) \mid f, g \in \mathrm{Ob}_{\mathbf{Ret}_b}, k \in \mathbf{C}(b, b), k = g \circ k \circ f\}$
3. $\mathrm{dom}(f, k, g) = f, \mathrm{cod}(f, k, g) = g$
4. $\mathrm{id}_f = (f, f, f)$
5. $(f, k, g) \circ (g', k', f) = (g', k \circ k', g)$

It is easy to verify that each arrow $(f,k,g) : f \to g$, defined in point 2, satisfy the following commutative diagram with two idempotent arrows f and g:

$$\begin{array}{ccc} b & \xrightarrow{\ f\ } & b \\ {\scriptstyle k}\downarrow & & \downarrow{\scriptstyle k} \\ b & \xleftarrow[\ g\ ]{} & b \end{array} \tag{5.7}$$

in the category **C** such that, from point 2 of Definition 24, $k = g \circ k \circ f$, so that we obtain that, from the idempotent property, $f = f \circ f$ and $g = g \circ g$:

$$g \circ k = g \circ (g \circ k \circ f) = g \circ k \circ f = k, \quad \text{and} \quad k \circ f = (g \circ k \circ f) \circ f = g \circ k \circ f = k,$$

i. e.,

$$g \circ k = k \circ f \tag{5.8}$$

so that the original commutative diagram (5.7) is equivalent to the more convenient (for application of the conceptual categorial symmetry) following commutative diagram:

$$\begin{array}{ccc} b & \xrightarrow{\ f\ } & b \\ {\scriptstyle k}\downarrow & & \downarrow{\scriptstyle k} \\ b & \xrightarrow[\ g\ ]{} & b \end{array} \quad \Leftrightarrow \quad \begin{array}{c} J(f) \\ \downarrow{\scriptstyle (k;k)} \\ J(g) \end{array} \quad \mapsto \quad \begin{array}{c} \widetilde{f} = T_e^0(J(f)) \\ \downarrow{\scriptstyle T_e^1(k;k)} \\ \widetilde{g} = T_e^0(J(g)) \end{array} \tag{5.9}$$

$$\text{in } \mathbf{C} \qquad\qquad \text{in } \mathbf{C} \downarrow \mathbf{C}$$

where the last vertical arrow is a candidate for a symmetry-extended category **C** (see Definition 14 in Section 3.4) with vertical composition of natural transformations (3.35), $\tau^{-1} \bullet \tau = \psi : F_{\text{st}} \xrightarrow{\cdot} S_{\text{nd}}$, with the functor $T_e : \mathbf{C} \downarrow \mathbf{C} \to \mathbf{C}$, which generates the conceptualized objects $\widetilde{f}$ and $\widetilde{g}$ from the arrows f and g. In this case, we obtain also the conceptual difference between the arrows f (the programs) and the consideration of them as "data" by using their form as conceptualized objects $\widetilde{f}$, also in the type-free languages.

Note that for each object b, at least its identity arrow $f = \text{id}_b : b \to b$ is idempotent, and that from the isomorphism $\tau_{J(\text{id}_b)} : b \simeq \widetilde{\text{id}_b}$ with its inverse arrow $\tau^{-1}_{J(\text{id}_b)} : \widetilde{\text{id}_b} \to b$ we have that $\tau_{J(\text{id}_b)} \circ \tau^{-1}_{J(\text{id}_b)} = \text{id}_{\widetilde{\text{id}_b}}$, i. e., $\widetilde{\text{id}}_b < b$, which satisfies the requirements for the symmetry-extended category **C**. So, the definition above is well-defined.

Thus, by using the category $\mathbf{Ret}_b$, we can define the subcategory of a conceptually-closed category $\mathbf{C}$, which objects are only the conceptualized objects obtained from the idempotent arrows.

Proposition 11. *Given a reflexive object b in a conceptually closed category $\mathbf{C}$, from Definition 7 in Section 3.3, let $\mathbf{ER}_b$ be its subcategory such that:*

1. *Each object is a conceptualized object $\widetilde{f} = T_e^0(J(f))$ obtained from an idempotent arrow $f : b \to b$ in $\mathbf{C}$.*
2. *Each arrow (f, k, g) in $\mathbf{Ret}_b$, such that $k = g \circ k \circ f$, corresponding to the arrow $(k;k) : J(f) \to J(g)$ in $\mathbf{C} \downarrow \mathbf{C}$, is represented in $\mathbf{ER}_b$ by the arrow $T_e^1(k;k) : \widetilde{f} \to \widetilde{g}$, where $\widetilde{f} = T_e^0(J(f))$ and $\widetilde{f} = T_e^0(J(f))$ are the conceptualized objects of $\mathbf{ER}_b$ obtained from the symmetry-closure functor $T_e : \mathbf{C} \downarrow \mathbf{C} \to \mathbf{C}$.*

Thus, the composition of objects $'*'$ (defined in CoCC category $\mathbf{C}$) of $\mathbf{ER}_b$ returns with an idempotent object, and hence $\mathbf{ER}_b$ is a category of idempotents on b.

Proof. Let us show that all objects of $\mathbf{ER}_b$ are *idempotent*. For the duality operator of arrow compositions, $B_\top = T_e^0 J$ (each symmetry-extended category is perfectly symmetric as well, and we can use the operation $*$ of corresponding composition of conceptualized objects in (3.5), we have that $B_\top(g \circ f) = B_\top(g) * B_\top(f)$), and hence we obtain

$$\begin{aligned}
\widetilde{f} * \widetilde{f} &= B_\top(f) * B_\top(f) \\
&= B_\top(f \circ f) \quad \text{from (3.5)} \\
&= B_\top(f) \quad \text{from idempotent arrow, } f = f \circ f \\
&= \widetilde{f}
\end{aligned}$$

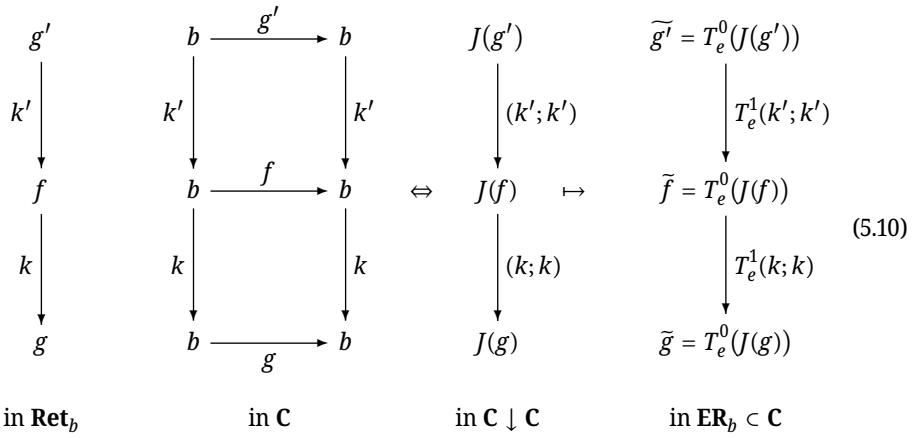
i. e., $\widetilde{f}$ is an idempotent conceptualized object (this is the duality property w. r. t. the composition of idempotent arrows $f : b \to b$). □

From Definition 24 of $\mathbf{Ret}_b$, it holds that the arrow k between two idempotent arrows $f, g : b \to b$, satisfies the property $k = g \circ k \circ f$, and hence the commutative diagram (5.7), so that the commutative diagram in (5.9) is represented by the arrow $(k;k) : J(f) \to J(g)$ in $\mathbf{C} \downarrow \mathbf{C}$, for which we have the following commutative diagram in $\mathbf{C}$:

$$\begin{array}{ccccc}
b = F_{\mathrm{st}}^0(J(f)) & \xrightarrow{\tau_{J(f)}} & \widetilde{f} & \xrightarrow{\tau^{-1}_{J(f)}} & b = S_{\mathrm{nd}}^0(J(f)) \\
k = \downarrow F_{\mathrm{st}}^1(k;k) & & \downarrow T_e^1(k;k) & & k = \downarrow S_{\mathrm{nd}}^1(k;k) \\
b = F_{\mathrm{st}}^0(J(g)) & \xrightarrow{\tau_{J(g)}} & \widetilde{g} & \xrightarrow{\tau^{-1}_{J(g)}} & b = S_{\mathrm{nd}}^0(J(g))
\end{array}$$

such that the middle vertical arrow is an arrow of $\mathbf{ER}_b$, $(\widetilde{f}, Te^1(k;k), \widetilde{g})$, analog to the arrow (f,k,g) in $\mathbf{Ret}_b$, from the fact that in both cases they represent the commutativity $k \circ f = g \circ k$, represented by the arrow $(k;k) : J(f) \to J(g)$ in category $\mathbf{C} \downarrow \mathbf{C}$.

The composition of the arrows in $\mathbf{Ret}_b$, $(f,k,g) \circ (g',k',f) = (g', k \circ k', g)$, given by point 5 of Definition 24, now is represented by this commutative diagram:



(5.10)

Consequently, the category $\mathbf{ER}_b$ is a well-defined substitution of the category of idempotents $\mathbf{Ret}_b$ by the category heaving conceptual categorial symmetry and clear distinction between (conceptualized) objects obtained from the idempotent arrows and the arrows also for the type-free languages. Next, we will show how $\mathbf{ER}_b$ can support the fixed-point operators.

5.3 Conceptually-closed CCC and its subcategories of idempotents

It was shown [11] that if a category $\mathbf{C}$ is a CCC with reflexive object b then, derived from it, the category of idempotents $\mathbf{Ret}_b$ in Definition 24 is a CCC as well, and hence can support the fixed-point operators. However, in this category of idempotents $\mathbf{Ret}_b$, both objects and arrows are the arrows of $\mathbf{C}$, with only difference that the objects of $\mathbf{Ret}_b$ are idempotent arrows of $\mathbf{C}$. Hence, the category $\mathbf{Ret}_b$ does not distinguish epistemologically the concepts of objects from the concept of arrows (e. g., the programs from their usage as data). It can be useful still to distinguish these two fundamental concepts at least epistemologically, and still be able to consider the type-free languages (where both "data" and 'programs" can be viewed as data). Typical example when also the programs are viewed as data are the fixed-point operators, where a program (expressed as an arrow of a given category) is considered as data for the fixed-point operator represented by another arrow in the same category.

Now, if we would like to use the categorial model for the fixed-point operators, such that in that category when a program f is viewed as such and represented by an arrow that elaborates the data of its domain and represent its results in its codomain object, from the case when f is considered as data for another programs (arrows in this category) and hence represented as an *object* of this category (the conceptualized object $\widetilde{f}$), then instead of the CCC category $\mathbf{Ret}_b$ we would like to use the category $\mathbf{ER}_b$ in which this epistemic difference is explicit and visible.

For such a possibility, for a given conceptually-closed CCC category $\mathbf{C}$ with a reflexive object b, from which we derive its subcategory $\mathbf{ER}_b$, we need first to demonstrate that also $\mathbf{ER}_b$ is CCC, and hence able to support the fixed-point operators.

Proposition 12. *Let a CCC category* $\mathbf{C}$ *be conceptually-closed by a functor* $T_e : \mathbf{C} \downarrow \mathbf{C} \to \mathbf{C}$, *and with a reflexive object* b. *Then its subcategory* $\mathbf{ER}_b$ *of idempotents on* b *is a CCC.*

Proof. Let b be a reflexive object in a CCC category $\mathbf{C}$, i. e., $(b \Rightarrow b) < b$, for the exponent object $b \Rightarrow b$ in this CCC, and it is well known that also the terminal object t and categorial product $b \times b$ in $\mathbf{C}$ are the retracts of b, via retract pairs:

- $t < b$ via in, out, so that $\mathrm{out} \circ \mathrm{in} = \mathrm{id}_t$;
- $b \times b < b$ via $\mathrm{in}_1, \mathrm{out}_1$, so that $\mathrm{out}_1 \circ \mathrm{in}_1 = \mathrm{id}_{b\times b}$;
- $(b \Rightarrow b) < b$ via $\mathrm{in}_2, \mathrm{out}_2$, so that $\mathrm{out}_2 \circ \mathrm{in}_2 = \mathrm{id}_{b\Rightarrow b}$;

for the product functors $\times : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$ and exponential functor $\Rightarrow : \mathbf{C}^{\mathrm{OP}} \times \mathbf{C} \to \mathbf{C}$.

Now we can define the terminal object, Cartesian product $\otimes$ and exponents in the category $\mathbf{ER}_b$, by

- Terminal object is conceptualized object $\widetilde{\mathrm{id}_t}$, for the idempotent identity arrow $\mathrm{id}_t : t \to t$, because for the conceptually closed category $\mathbf{C}$, we have the isomorphic arrow $\varphi(t) : \widetilde{\mathrm{id}_t} \to t$, from the natural isomorphism (3.13), with its inverse $\varphi^{-1}(t) : t \to \widetilde{\mathrm{id}_t}$, and hence we have the retract $\widetilde{\mathrm{id}_t} < b$ via $\mathrm{in}_0, \mathrm{out}_0$, such that $\mathrm{in}_0 = \mathrm{in} \circ \varphi(t)$ and $\mathrm{out}_0 = \varphi^{-1}(t) \circ \mathrm{out}$ (in fact, it holds that $\mathrm{out}_0 \circ \mathrm{in}_0 = (\varphi^{-1}(t) \circ \mathrm{out}) \circ (\mathrm{in} \circ \varphi(t)) = \varphi^{-1}(t) \circ \varphi(t) = \mathrm{id}_{\widetilde{\mathrm{id}_t}}$).

  Thus, $\widetilde{\mathrm{id}_t} < b$ via $\mathrm{in}_0, \mathrm{out}_0$.
- For any two *idempotent* arrows $f, g : b \to b$ and their conceptualized objects $\widetilde{f} = T_e^0(J(f))$ and $\widetilde{g} = T_e^0(J(g))$ of $\mathbf{ER}_b$, we define their product by $\widetilde{f} \otimes \widetilde{g} = \widetilde{h}$ where $h : b \to b$ is the following arrow in $\mathbf{C}$:

$$h \triangleq \mathrm{in}_1 \circ (f \times g) \circ \mathrm{out}_1 \tag{5.11}$$

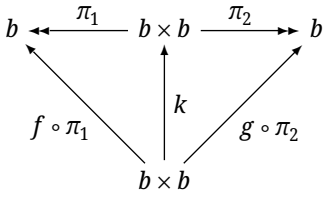
  equal to idempotent arrow used to define the product of f and g in $\mathbf{Ret}_b$. In fact, $h \circ h = \mathrm{in}_1 \circ (f \times g) \circ \mathrm{out}_1 \circ \mathrm{in}_1 \circ (f \times g) \circ \mathrm{out}_1 = \mathrm{in}_1 \circ (f \times g) \circ (f \times g) \circ \mathrm{out}_1 = \mathrm{in}_1 \circ ((f \circ f) \times (g \circ g)) \circ \mathrm{out}_1 = \mathrm{in}_1 \circ (f \times g) \circ \mathrm{out}_1 = h$.
- For any two *idempotent* arrows $f, g : b \to b$ and their conceptualized objects $\widetilde{f} = T_e^0(J(f))$ and $\widetilde{g} = T_e^0(J(g))$ of $\mathbf{ER}_b$, we define their exponent by $\widetilde{g^f}$ where $g^f : b \to b$ is the following arrow in $\mathbf{C}$:

$$g^f \triangleq \mathrm{in}_2 \circ \Lambda(h_0) \circ \mathrm{out}_2, \quad \text{where } h_0 = g \circ \mathrm{eval}_{b,b} \circ (\mathrm{id}_{b \Rightarrow b} \times f) : (b \Rightarrow b) \times b \to b \tag{5.12}$$

equal to idempotent arrow used to define the exponent of f and g in $\mathbf{Ret}_b$. In fact,

$$\begin{aligned}
\Lambda(h_0) \circ \Lambda(h_0) &= \Lambda(g \circ \mathrm{eval}_{b,b} \circ (\mathrm{id} \times f)) \circ \Lambda(g \circ \mathrm{eval}_{b,b} \circ (\mathrm{id} \times f)) \\
&= \Lambda(g \circ \mathrm{eval}_{b,b} \circ (\mathrm{id} \times f) \circ \Lambda(g \circ \mathrm{eval}_{b,b} \circ (\mathrm{id} \times f)) \times \mathrm{id}) \\
&\quad \text{from } (\beta) \text{ reduction} \\
&= \Lambda(g \circ \mathrm{eval}_{b,b} \circ \Lambda(g \circ \mathrm{eval}_{b,b} \circ (\mathrm{id} \times f)) \times \mathrm{id} \circ (\mathrm{id} \times f)) \\
&= \Lambda(g \circ (g \circ \mathrm{eval}_{b,b} \circ (\mathrm{id} \times f)) \circ (\mathrm{id} \times f)) \\
&= \Lambda((g \circ g) \circ \mathrm{eval}_{b,b} \circ (\mathrm{id} \times (f \circ f))) = \Lambda(h_0), \\
&\quad \text{so that } g^f \text{ is idempotent,} \\
g^f \circ g^f &= \mathrm{in}_2 \circ \Lambda(h_0) \circ \mathrm{out}_2 \circ \mathrm{in}_2 \circ \Lambda(h_0) \circ \mathrm{out}_2 = \mathrm{in}_2 \circ \Lambda(h_0) \circ \Lambda(h_0) \circ \mathrm{out}_2 = g^f .
\end{aligned}$$

Note that in any CCC category $\mathbf{C}$, such that the following product limit in $\mathbf{C}$ commutes:



where, for the product projections π_1 and π_2, the unique vertical arrow for this commutative diagram define the pairing operation $\langle _, _ \rangle$ in a CCC,

$$k = \langle f \circ \pi_1, g \circ \pi_2 \rangle = f \times g \tag{5.13}$$

with the compositional properties

$$\langle f \circ m, g \circ m \rangle = \langle f, g \rangle \circ m, \quad (m \times n) \circ \langle f, g \rangle = \langle m \circ f, n \circ g \rangle, \tag{5.14}$$

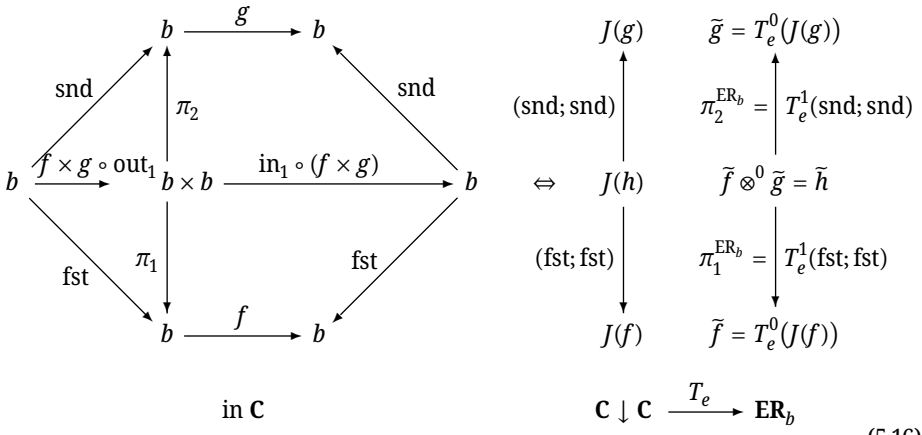
and hence it holds that

$$f \circ \pi_1 = \pi_1 \circ (f \times g), \quad g \circ \pi_2 = \pi_2 \circ (f \times g). \tag{5.15}$$

Thus we have to show that these definitions of products and exponents in $\mathbf{ER}_b$ are well-defined:

1. Proof for the *products* $\otimes$ in $\mathbf{ER}_b$:
Let us consider the following commutative diagram in CCC conceptually closed category $\mathbf{C}$, for three idempotent arrows: the top and bottom horizontal arrows g, f and middle horizontal arrow given by (5.11),

$$h = (\mathrm{in}_1 \circ (f \times g)) \circ (f \times g) \circ \mathrm{out}_1 = \mathrm{in}_1 \circ (f \times g) \circ \mathrm{out}_1$$



(5.16)

Thus, the two new arrows in diagram (5.16),

$$\mathrm{fst} = \pi_1 \circ (f \times g) \circ \mathrm{out}_1, \quad \mathrm{snd} = \pi_2 \circ (f \times g) \circ \circ \mathrm{out}_1 \tag{5.17}$$

satisfy the requirement for the arrows of $\mathbf{Ret}_b$ (see in the diagram above the two arrows in $\mathbf{C} \downarrow \mathbf{C}$), i. e.,

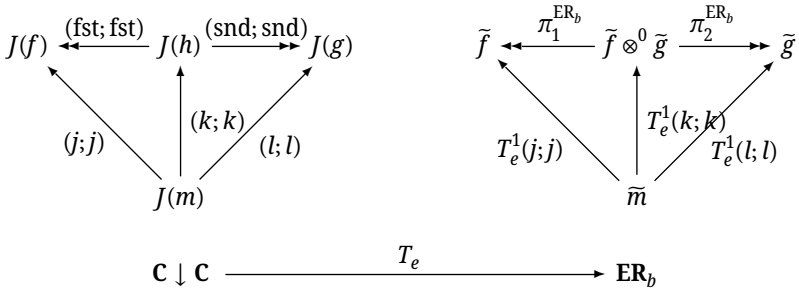
$$\begin{aligned} f \circ \mathrm{fst} \circ h &= f \circ (\pi_1 \circ (f \times g) \circ \mathrm{out}_1) \circ \mathrm{in}_1 \circ (f \times g) \circ \mathrm{out}_1 \\ &= f \circ \pi_1 \circ (f \times g) \circ \mathrm{out}_1 \\ &= \pi_1 \circ (f \times g) \circ (f \times g) \circ \mathrm{out}_1 \quad \text{from (5.15)} \\ &= \pi_1 \circ (f \times g) \circ \mathrm{out}_1 \\ &= \mathrm{fst} \end{aligned}$$

and analogously, $\mathrm{snd} = g \circ \mathrm{snd} \circ h$. It is easy to verify, from (5.13), that also $f \circ \pi_1 = \mathrm{fst} \circ \mathrm{in}_1 \circ (f \times g)$ and $g \circ \pi_2 = \mathrm{snd} \circ \mathrm{in}_1 \circ (f \times g)$, so that diagram in (5.16) commutes.

Consequently, the first and second projections of the (limit) products in $\mathbf{ER}_b$ are defined by symmetry-closure functor (see (3.12), and hence the "isomorphism" $T_e^1(f; f) \cong f$),

$$\pi_1^{\mathrm{ER}_b} \triangleq T_e^1(\mathrm{fst}; \mathrm{fst}), \quad \pi_2^{\mathrm{ER}_b} \triangleq T_e^1(\mathrm{snd}; \mathrm{snd}) \tag{5.18}$$

such that for any two arrows $(j; j) : J(m) \to J(f)$ and $(l; l) : J(m) \to J(g)$, for a given *idempotent* arrow $m : b \to b$ of $\mathbf{C}$, from the diagram (5.16), the following limit (product) diagrams commute:



where $k = \mathrm{in}_1 \circ \langle j, l \rangle$ for the pairing operation $\langle , \rangle$ in $\mathbf{C}$. From the requirement for the arrows j and l in $\mathbf{Ret}_b$, from the left-hand side of diagram above it holds that $j = f \circ j \circ m$ and $l = g \circ l \circ m$, and hence for the arrow k, we obtain

$$\begin{aligned} h \circ k \circ m &= (\mathrm{in}_1 \circ (f \times g) \circ \mathrm{out}_1) \circ (\mathrm{in}_1 \circ \langle j, l \rangle) \circ m \\ &= \mathrm{in}_1 \circ (f \times g) \circ \langle j \circ m, l \circ m \rangle \\ &= \mathrm{in}_1 \circ \langle f \circ j \circ m, g \circ l \circ m \rangle \\ &= \mathrm{in}_1 \circ \langle j, l \rangle = k, \end{aligned}$$

and hence also k is well-defined arrow w. r. t. the category $\mathbf{Rev}_b$.

That this definition is correct for the commutative diagram above, we can verify, from $j = f \circ j \circ m$ and $l = g \circ l \circ m$, by

$$\begin{aligned} \mathrm{fst} \circ k &= (\pi_1 \circ (f \times g) \circ \mathrm{out}_1) \circ (\mathrm{in}_1 \circ \langle j, l \rangle) = \pi_1 \circ \langle f \circ j, g \circ l \rangle = \pi_1 \circ \langle j, l \rangle = j, \\ \mathrm{snd} \circ k &= (\pi_2 \circ (f \times g) \circ \mathrm{out}_1) \circ (\mathrm{in}_1 \circ \langle j, l \rangle) = \pi_2 \circ \langle f \circ j, g \circ l \rangle = \pi_2 \circ \langle j, l \rangle = l, \end{aligned}$$

so that the pairing operation $\langle , \rangle^{\mathrm{ER}_b}$ in $\mathbf{ER}_b$ is defined by

$$\langle T_e^1(j; j), T_e^1(l; l) \rangle^{\mathrm{ER}_b} \triangleq T_e^1(k; k) = T_e^1(\mathrm{in}_1 \circ \langle j, l \rangle; \mathrm{in}_1 \circ \langle j, l \rangle) \tag{5.19}$$

Let us now define also the arrow component of the product functor $\otimes$, and consider any two arrows $T_e^1(h_i; h_i) : \widetilde{f}_i \to \widetilde{g}_i$, for $i = 1, 2$, in $\mathbf{ER}_b$, so that it is satisfied,

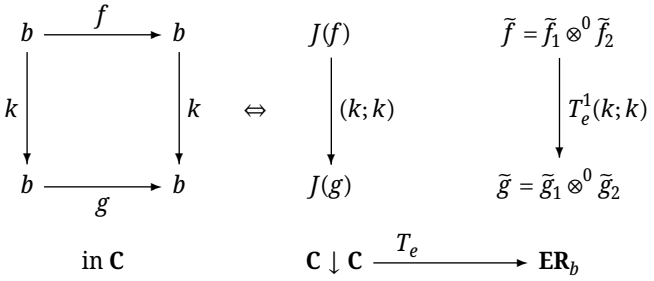
$$g_i \circ h_i = h_i \circ f_i, \quad i = 1, 2, \tag{5.20}$$

then for two idempotent arrows $f = \mathrm{in}_1 \circ (f_1 \times f_2) \circ \mathrm{out}_1$, $g = \mathrm{in}_1 \circ (g_1 \times g_2) \circ \mathrm{out}_1$, and the arrow $k = \mathrm{in}_1 \circ ((h_1 \circ f_1) \times (h_2 \circ f_2)) \circ \mathrm{out}_1$, we obtain

$$\begin{aligned} g \circ k &= (\mathrm{in}_1 \circ (g_1 \times g_2) \circ \mathrm{out}_1) \circ (\mathrm{in}_1 \circ ((h_1 \circ f_1) \times (h_2 \circ f_2)) \circ \mathrm{out}_1) \\ &= \mathrm{in}_1 \circ ((g_1 \circ h_1 \circ f_1) \times (g_2 \circ h_2 \circ f_2)) \circ \mathrm{out}_1 \\ &= \mathrm{in}_1 \circ ((h_1 \circ f_1 \circ f_1) \times (h_2 \circ f_2 \circ f_2)) \circ \mathrm{out}_1 \quad \text{from (5.20)} \\ &= \mathrm{in}_1 \circ ((h_1 \circ f_1) \times (h_2 \circ f_2)) \circ (f_1 \times f_2) \circ \mathrm{out}_1 \end{aligned}$$

$$
\begin{aligned}
&= \mathrm{in}_1 \circ ((h_1 \circ f_1) \times (h_2 \circ f_2)) \circ \mathrm{out}_1 \circ \mathrm{in}_1 \circ (f_1 \times f_2) \circ \mathrm{out}_1 \\
&= k \circ f
\end{aligned}
$$

and hence, the following diagram commutes:



(5.21)

Thus, from the last vertical arrow in $\mathbf{ER}_b$, we obtain the definition of the product of two arrows in $\mathbf{ER}_b$, $T_e^1(h_i; h_i) : \widetilde{f}_i \to \widetilde{g}_i$, for $i = 1, 2$,

$$
T_e^1(h_1; h_1) \otimes^1 T_e^1(h_2; h_2) \triangleq T_e^1(k; k) : \widetilde{f}_1 \otimes^0 \widetilde{f}_2 \to \widetilde{g}_1 \otimes^0 \widetilde{g}_2 \tag{5.22}
$$

where $k = \mathrm{in}_1 \circ ((h_1 \circ f_1) \times (h_2 \circ f_2)) \circ \mathrm{out}_1$.

2. Proof for the *exponents* in $\mathbf{ER}_b$:
We have defined the exponent objects in $\mathbf{ER}_b$ by conceptualized object $\widetilde{g^f}$ obtained from the idempotent arrow in (5.12), $g^f = \mathrm{in}_2 \circ \Lambda(h_0) \circ \mathrm{out}_2$, where both $f, g : b \to b$ in CCC category $\mathbf{C}$ are idempotent arrows.

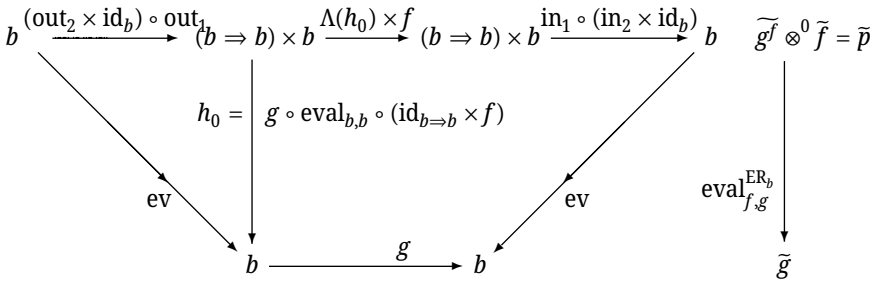
Thus, in order to satisfy the requirements for an exponent object, we need to have in $\mathbf{ER}_b$ also the evaluation arrow $\mathrm{eval}_{f,g}^{\mathrm{ER}_b} : \widetilde{g^f} \otimes^0 \widetilde{f} \to \widetilde{g}$, where from (5.11) the product $\widetilde{g^f} \otimes^0 \widetilde{f} = \widetilde{p}$ is obtained as conceptualized object of the arrow

$$
p = \mathrm{in}_1 \circ (g^f \times f) \circ \mathrm{out}_1 = \mathrm{in}_1 \circ ((\mathrm{in}_2 \circ \Lambda(h_0) \circ \mathrm{out}_2) \times f) \circ \mathrm{out}_1 \tag{5.23}
$$

and we has to show that it is idempotent arrow in order to have that $\widetilde{p}$ is a well-defined object of $\mathbf{ER}_b$, as follows:

$$
\begin{aligned}
p \circ p &= (\mathrm{in}_1 \circ (g^f \times f) \circ \mathrm{out}_1) \circ (\mathrm{in}_1 \circ (g^f \times f) \circ \mathrm{out}_1) \\
&= \mathrm{in}_1 \circ (g^f \times f) \circ (g^f \times f) \circ \mathrm{out}_1 \\
&= \mathrm{in}_1 \circ ((g^f \circ g^f) \times (f \circ f)) \circ \mathrm{out}_1 \\
&= \mathrm{in}_1 \circ (g^f \times f) \circ \mathrm{out}_1 \\
&= p,
\end{aligned}
$$

from the fact that both g^f and f are idempotent arrows. This idempotent arrow $p : b \to b$ is represented as the upper horizontal arrow in the following commutative diagram in $\mathbf{C}$:



(5.24)

where the left commutative diagram defines the arrow $\mathrm{ev} : b \to b$ of $\mathbf{C}$ by

$$\mathrm{ev} \triangleq h_0 \circ (\mathrm{out}_2 \times \mathrm{id}_b) \circ \mathrm{out}_1 = g \circ \mathrm{eval}_{b,b} \circ (\mathrm{out}_2 \times f) \circ \mathrm{out}_1. \tag{5.25}$$

So, it is enough to prove that the right-hand side of this diagram commutes as well,

$$\begin{aligned}
&\mathrm{ev} \circ \mathrm{in}_1 \circ (\mathrm{in}_2 \times \mathrm{id}) \circ (\Lambda(h_0) \times f) \\
&\quad = (g \circ \mathrm{eval}_{b,b} \circ (\mathrm{out}_2 \times f) \circ \mathrm{out}_1) \circ \mathrm{in}_1 \circ (\mathrm{in}_2 \times \mathrm{id}) \circ (\Lambda(h_0) \times f) \\
&\quad = g \circ \mathrm{eval}_{b,b} \circ (\mathrm{out}_2 \times f) \circ (\mathrm{in}_2 \times \mathrm{id}) \circ (\Lambda(h_0) \times f) \\
&\quad = g \circ \mathrm{eval}_{b,b} \circ (\mathrm{id} \times f) \circ (\Lambda(h_0) \times f) = g \circ \mathrm{eval}_{b,b} \circ (\Lambda(h_0) \times (f \circ f)) \\
&\quad = g \circ (\mathrm{eval}_{b,b} \circ (\Lambda(h_0) \times \mathrm{id})) \circ (\mathrm{id} \times (f \circ f)) \\
&\quad = g \circ h_0 \circ (\mathrm{id} \times (f \circ f)) \quad \text{by } (\beta) \text{ reduction} \\
&\quad = g \circ (g \circ \mathrm{eval}_{b,b} \circ (\mathrm{id}_{b\Rightarrow b} \times f)) \circ (\mathrm{id} \times (f \circ f)) \quad \text{by (5.12)} \\
&= g \circ (g \circ \mathrm{eval}_{b,b} \circ (\mathrm{id}_{b\Rightarrow b} \times f)) \quad \text{by the idempotent arrow } f \\
&= g \circ h_0
\end{aligned}$$

i. e., also this part commutes, and hence, the whole diagram in (5.24) commutes, i.e, $\mathrm{ev} \circ p = g \circ \mathrm{ev}$, and hence, from fact that g is idempotent,

$$g \circ (\mathrm{ev} \circ p) = g \circ (g \circ \mathrm{ev}) = g \circ \mathrm{ev} = \mathrm{ev},$$

from the definition of ev in (5.25) so that (p, ev, g) is a well-defined arrow in $\mathbf{Ret}_b$, and hence we can use the arrow $(\mathrm{ev}; \mathrm{ev}) : J(p) \to J(g)$ in the category $\mathbf{C} \downarrow \mathbf{C}$, and by using the symmetry-closure functor T_e, we obtain that the evaluation arrow in $\mathbf{ER}_b$ represented by the vertical arrow in diagram above, such that

$$\mathrm{eval}_{f,g}^{\mathrm{ER}_b} \triangleq T_e^1(\mathrm{ev}; \mathrm{ev}) : \widetilde{g^f} \otimes \tilde{f} \to \tilde{g}, \tag{5.26}$$

and from (3.12), with the "isomorphism" $T_e^1(\mathrm{ev}; \mathrm{ev}) \cong \mathrm{ev}$, where the exponent object is

$$\widetilde{g^f} = T_e^0(J(\mathrm{in}_2 \circ \Lambda(h_0) \circ \mathrm{out}_2)) = T_e^0(J(\mathrm{in}_2 \circ \Lambda(g \circ \mathrm{eval}_{b,b} \circ (\mathrm{id}_{b\Rightarrow b} \times f)) \circ \mathrm{out}_2)),$$

and hence from (5.25),

$$\mathrm{eval}^{\mathbf{ER}_b}_{f,g} = T^1_e(g \circ \mathrm{eval}_{b,b} \circ (\mathrm{out}_2 \times f) \circ \mathrm{out}_1;\ g \circ \mathrm{eval}_{b,b} \circ (\mathrm{out}_2 \times f) \circ \mathrm{out}_1).$$

Moreover, for a given idempotent arrow m and any arrow $(k;k) : J(\mathrm{in}_1 \circ (m \times f) \circ \mathrm{out}_1) \to J(g)$ of $\mathbf{C} \downarrow \mathbf{C}$, such that $(\mathrm{in}_1 \circ (m \times f) \circ \mathrm{out}_1, k, g)$ is an arrow in $\mathbf{Ret}_b$, i. e.,

$$k = g \circ k \circ (\mathrm{in}_1 \circ (m \times f) \circ \mathrm{out}_1), \tag{5.27}$$

and hence with the commutativity

$$g \circ k = k \circ (\mathrm{in}_1 \circ (m \times f) \circ \mathrm{out}_1) \tag{5.28}$$

where $m : b \to b$ is an idempotent arrow in $\mathbf{C}$, by application of the functor T_e to the arrow $(k;k)$, we obtain the following arrow in $\mathbf{ER}_b$:

$$T^1_e(k;k) : \widetilde{m} \otimes^0 \widetilde{f} \to \widetilde{g} \tag{5.29}$$

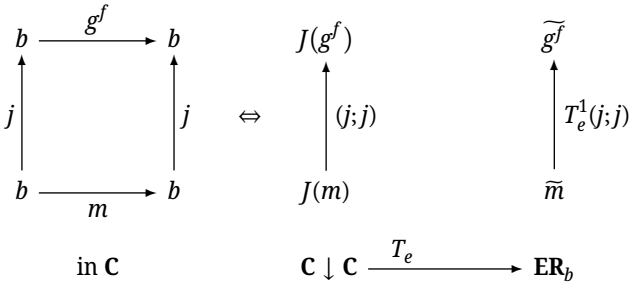
Let us define the arrow

$$j \triangleq \mathrm{in}_2 \circ \Lambda(k \circ \mathrm{in}_1) : b \to b \tag{5.30}$$

and we can show that the arrow (g^f, j, m) in $\mathbf{Ret}_b$ is well-defined, i. e.,

$$\begin{aligned}
g^f \circ j \circ m &= (\mathrm{in}_2 \circ \Lambda(g \circ \mathrm{eval}_{b,b} \circ (\mathrm{id} \times f)) \circ \mathrm{out}_2) \circ (\mathrm{in}_2 \circ \Lambda(k \circ \mathrm{in}_1)) \circ m \\
&= \mathrm{in}_2 \circ \Lambda(g \circ \mathrm{eval}_{b,b} \circ (\mathrm{id} \times f)) \circ \Lambda(k \circ \mathrm{in}_1 \circ (m \times \mathrm{id})) \quad \text{by Curry} \\
&= \mathrm{in}_2 \circ \Lambda(g \circ \mathrm{eval}_{b,b} \circ (\mathrm{id} \times f) \circ (\Lambda(k \circ \mathrm{in}_1 \circ (m \times \mathrm{id})) \times \mathrm{id})) \quad \text{by Curry} \\
&= \mathrm{in}_2 \circ \Lambda(g \circ \mathrm{eval}_{b,b} \circ (\Lambda(k \circ \mathrm{in}_1 \circ (m \times \mathrm{id})) \times \mathrm{id}) \circ (\mathrm{id} \times f)) \\
&= \mathrm{in}_2 \circ \Lambda(g \circ k \circ \mathrm{in}_1 \circ (m \times \mathrm{id}) \circ (\mathrm{id} \times f)) \quad \text{by } (\beta) \text{ reduction} \\
&= \mathrm{in}_2 \circ \Lambda(g \circ k \circ \mathrm{in}_1 \circ (m \times f)) \\
&= \mathrm{in}_2 \circ \Lambda((g \circ k \circ \mathrm{in}_1 \circ (m \times f) \circ \mathrm{out}_1) \circ \mathrm{in}_1) \\
&= \mathrm{in}_2 \circ \Lambda(k \circ \mathrm{in}_1) \quad \text{by (5.30)} \\
&= j,
\end{aligned}$$

and hence, the following diagram commutes:



(5.31)

Thus, based on the last vertical arrow in diagram (5.31), we can define the abstraction operation Λ^{RE_b} in $\mathbf{ER}_b$ for the arrow in (5.29),

$$\Lambda^{\mathrm{RE}_b}(T_e^1(k;k)) \triangleq T_e^1(j;j) : \widetilde{m} \to \widetilde{g^f} \quad \text{with } j = \mathrm{in}_2 \circ \Lambda(k \circ \mathrm{in}_1) \tag{5.32}$$

from (3.12), with the isomorphism $T_e^1(j;j) \cong j$, in our candidate exponential diagram in $\mathbf{ER}_b$,

$$\begin{array}{ccc} \widetilde{g^f} \otimes^0 \widetilde{f} & \xrightarrow{\mathrm{eval}_{f,g}^{\mathrm{RE}_b}} & \widetilde{g} \\ \uparrow \Lambda^{\mathrm{RE}_b}(T_e^1(k;k)) \otimes^1 \mathrm{id}_{\widetilde{f}} & \nearrow_{T_e^1(k;k)} & \\ \widetilde{m} \otimes^0 \widetilde{f} & & \end{array} \tag{5.33}$$

where the identity arrow $\mathrm{id}_{\widetilde{f}} = T_e^1(\mathrm{id}_b; \mathrm{id}_b)$, so that from definition of the product of the arrows in (5.22), we define $T_e^1(l;l) \triangleq \Lambda^{\mathrm{RE}_b}(T_e^1(k;k)) \otimes^1 \mathrm{id}_{\widetilde{f}} = T_e^1(j;j) \otimes^1 T_e^1(\mathrm{id}_b; \mathrm{id}_b)$, i. e.,

$$l \triangleq \mathrm{in}_1 \circ ((j \circ m) \times (\mathrm{id}_b \circ f)) \circ \mathrm{out}_1, \tag{5.34}$$

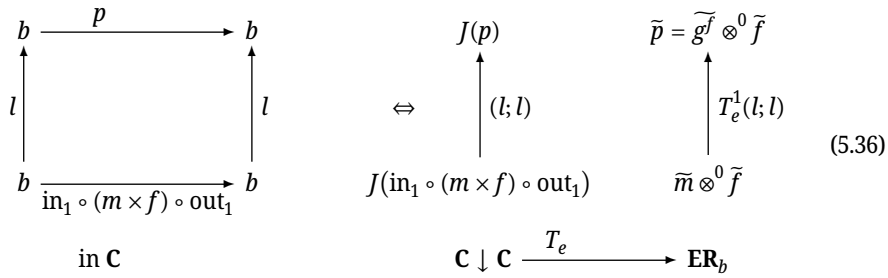
and hence, from (5.30), we obtain this detailed expression,

$$l = \mathrm{in}_1 \circ ((\mathrm{in}_2 \circ \Lambda(k \circ \mathrm{in}_1) \circ m) \times f) \circ \mathrm{out}_1. \tag{5.35}$$

Let us show that it corresponds to the arrow $(\mathrm{in}_1 \circ (m \times f) \circ \mathrm{out}_1, l, p)$ in $\mathbf{Ret}_b$, i. e.,

$$\begin{aligned} & p \circ l \circ \mathrm{in}_1 \circ (m \times f) \circ \mathrm{out}_1 \\ & \quad = p \circ \mathrm{in}_1 \circ ((j \circ m) \times f) \circ \mathrm{out}_1 \circ \mathrm{in}_1 \circ (m \times f) \circ \mathrm{out}_1 \quad \text{by (5.34)} \\ & \quad = p \circ \mathrm{in}_1 \circ ((j \circ m \circ m) \times (f \circ f)) \circ \mathrm{out}_1 \\ & \quad = p \circ \mathrm{in}_1 \circ ((j \circ m) \times f) \circ \mathrm{out}_1 \quad \text{by the idempotent arrows } m \text{ and } f \\ & \quad = (\mathrm{in}_1 \circ (g^f \times f) \circ \mathrm{out}_1) \circ \mathrm{in}_1 \circ ((j \circ m) \times f) \circ \mathrm{out}_1 \quad \text{by (5.23)} \\ & \quad = \mathrm{in}_1 \circ (((g^f \circ j) \circ m) \times f) \circ \mathrm{out}_1 \\ & \quad = \mathrm{in}_1 \circ (((j \circ m) \circ m) \times f) \circ \mathrm{out}_1 \quad \text{by the commutative diagram (5.31)} \\ & \quad = \mathrm{in}_1 \circ ((j \circ m) \times f) \circ \mathrm{out}_1 \\ & \quad = l \quad \text{by (5.34),} \end{aligned}$$

and hence, the following diagram commutes:

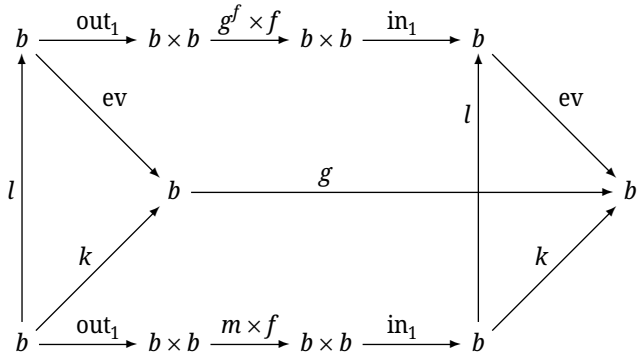


(5.36)

From the fact that from (5.26), $\text{eval}^{\text{RE}_b}_{f,g} = T^1_1(\text{ev}; \text{ev})$, let us show that the exponential diagram (5.33) commutes. In fact, we have from (5.25) and (5.35),

$$\begin{aligned}
\text{ev} \circ l &= (g \circ \text{eval}_{b,b} \circ (\text{out}_2 \times f) \circ \text{out}_1) \circ (\text{in}_1 \circ ((\text{in}_2 \circ \Lambda(k \circ \text{in}_1) \circ m) \times f) \circ \text{out}_1) \\
&= g \circ \text{eval}_{b,b} \circ ((\text{out}_2 \circ \text{in}_2 \circ \Lambda(k \circ \text{in}_1) \circ m) \times (f \circ f)) \circ \text{out}_1 \\
&= g \circ \text{eval}_{b,b} \circ ((\Lambda(k \circ \text{in}_1) \circ m) \times f) \circ \text{out}_1 \\
&= g \circ \text{eval}_{b,b} \circ ((\Lambda(k \circ \text{in}_1) \times \text{id}) \circ (m \times f)) \circ \text{out}_1 \\
&= g \circ k \circ \text{in}_1 \circ (m \times f) \circ \text{out}_1 \quad \text{by } (\beta) \text{ reduction} \\
&= k \quad \text{from (5.27).}
\end{aligned}$$

Let us verify that the exponential diagram (5.33) in $\mathbf{ER}_b$ is directly derived from the following commutative diagram in $\mathbf{C}$, where the top horizontal arrow is $p = \text{in}_1 \circ (g^f \times f) \circ \text{out}_1$ from (5.23), where from (5.12) $g^f = \text{in}_2 \circ \Lambda(h_0) \circ \text{out}_2$, with $h_0 = g \circ \text{eval}_{b,b} \circ (\text{id}_{b \Rightarrow b} \times f)$, such that the conceptualized object is $\widetilde{p} = \widetilde{g^f} \otimes^0 \widetilde{f}$, while the conceptualized object $\widetilde{m} \otimes \widetilde{f}$, for a given idempotent arrow $m : b \to b$ is obtained from the bottom horizontal arrow:



(5.37)

The upper rhombus is the commutative diagram (5.24) with $\text{ev} \circ p = g \circ \text{ev}$. So, for the arrow $k : b \to b$ for which we assume the commutativity of the lower rhombus, and the commutativity of the left/right small triangles, $k = \text{ev} \circ l$ is shown above in this diagram.

This completes the proof that $\mathbf{ER}_b$ is a CCC. □

5.4 Internal categorial symmetry and fixed-point operators

In this section, we will provide the final theorem, which uses the definition of the fixed-point operators fix_b in a CCC category $\mathbf{C}$, specified in Proposition 10 in Section 5.2 with the commutative diagram (5.2), i. e.,

$$\text{fix}_b = \text{eval}_{b,b} \circ (\text{id}_{b\Rightarrow b} \times \text{fix}_b) : (b \Rightarrow b) \to b. \tag{5.38}$$

However, it is valid *only* for the reflexive objects b in C, and hence, $\mathbf{C}$ has no fixed-point operators because not all objects of $\mathbf{C}$ are reflexive.

Consequently, in order to obtain a category, which has fixed-point operators (from Proposition 10), we can use the idempotent categories $\mathbf{ER}_b$ for which we have demonstrated that are CCC as well, because in $\mathbf{ER}_b$ every object has the fixed-point operators, as we can demonstrate by the following theorem.

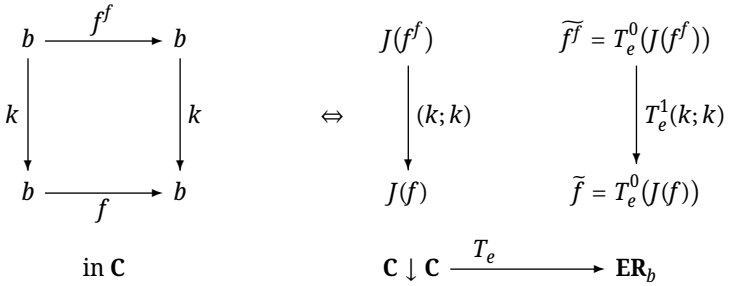
Theorem 9. *Every object $\widetilde{f}$ in $\mathbf{ER}_b$ has a fixed-point operator* Fix_f^{ER}.

Proof. We recall that each object $\widetilde{f}$ of $\mathbf{ER}_b$ is a conceptualized object equal to $T_e^0(J(f))$ where $f : b \to b$ is an idempotent morphism of the conceptually closed CCC category $\mathbf{C}$ by the symmetry-closure functor $T_e : \mathbf{C} \downarrow \mathbf{C} \to \mathbf{C}$.

Let $\text{fix}_b : (b \Rightarrow b) \to b$ be a fixed-point operator for the reflexive object b in $\mathbf{C}$. We can define the following morphism in the CCC category $\mathbf{C}$:

$$k \triangleq \text{fix}_b \circ \text{out}_2 \circ f^f : b \to b \tag{5.39}$$

where from (5.12), $f^f = \text{in}_2 \circ \Lambda(h_0) \circ \text{out}_2 : b \to b$, with $h_0 = f \circ \text{eval}_{b,b} \circ (\text{id}_{b\Rightarrow b} \times f) : (b \Rightarrow b) \times b \to b$, and we can verify that it is good definition (as required by the arrows in $\mathbf{Ret}_b$, i. e., that holds $k = f \circ k \circ f^f$, so that the following diagram in the conceptually-closed CCC category $\mathbf{C}$ commutes:



(5.40)

where the fixed-point operator Fix_f^{ER} for the object $\widetilde{f}$ in $\mathbf{ER}_b$ corresponds to the right-hand side vertical arrow in the diagram above, i. e.,

$$\text{Fix}_f^{\text{ER}_b} \triangleq T_e^1(k;k) = T_e^1(\text{fix}_b \circ \text{out}_2 \circ f^f ; \text{fix}_b \circ \text{out}_2 \circ f^f) : \widetilde{f^f} \to \widetilde{f}. \tag{5.41}$$

Let us show that the diagram above commutes indeed, and hence, we define an auxiliary arrow $h_1 = \mathrm{eval}_{b,b} \circ \langle \mathrm{id}_{b\Rightarrow b}, \mathrm{in}_2 \rangle \circ \mathrm{out}_2 : b \to b$ and also

$$h = \mathrm{eval}_{b,b} \circ (\mathrm{id}_{b\Rightarrow b} \times f) \circ (\mathrm{out}_2 \times h_1) : b \times b \to b,$$

so that we obtain

$$\begin{aligned}
k &= \mathrm{fix}_b \circ \mathrm{out}_2 \circ f^f \\
&= \mathrm{eval} \circ \langle \mathrm{id}, \mathrm{in}_2 \rangle \circ \Lambda(\mathrm{eval} \circ (\mathrm{id} \times h_1)) \circ \mathrm{out}_2 \circ f^f \quad \text{by substitution of (5.3)} \\
&= \mathrm{eval} \circ \langle \mathrm{id}, \mathrm{in}_2 \rangle \circ \Lambda(\mathrm{eval} \circ (\mathrm{id} \times h_1) \circ ((\mathrm{out}_2 \circ f^f) \times \mathrm{id})) \quad \text{by Curry} \\
&= \mathrm{eval} \circ \langle \mathrm{id}, \mathrm{in}_2 \rangle \circ \Lambda(\mathrm{eval} \circ ((\mathrm{out}_2 \circ f^f) \times h_1)) \\
&= \mathrm{eval} \circ \langle \mathrm{id}, \mathrm{in}_2 \rangle \circ \Lambda(\mathrm{eval} \circ (\Lambda(f \circ \mathrm{eval}_{b,b} \circ (\mathrm{id} \times f)) \times \mathrm{id}) \circ (\mathrm{out}_2 \times h_1)) \quad \text{by } f^f \\
&= \mathrm{eval} \circ \langle \mathrm{id}, \mathrm{in}_2 \rangle \circ \Lambda(f \circ \mathrm{eval}_{b,b} \circ (\mathrm{id} \times f) \circ (\mathrm{out}_2 \times h_1)) \quad \text{by } (\beta) \\
&= \mathrm{eval} \circ \langle \mathrm{id}, \mathrm{in}_2 \rangle \circ \Lambda(f \circ h) \\
&= \mathrm{eval} \circ \langle \Lambda(f \circ h), \mathrm{in}_2 \circ \Lambda(f \circ h) \rangle \quad \text{by (5.14)} \\
&= \mathrm{eval} \circ (\Lambda(f \circ h) \times \mathrm{id}) \circ \langle \mathrm{id}, \mathrm{in}_2 \circ \Lambda(f \circ h) \rangle \quad \text{by (5.14)} \\
&= f \circ h \circ \langle \mathrm{id}, \mathrm{in}_2 \circ \Lambda(f \circ h) \rangle \quad \text{by } (\beta)
\end{aligned}$$

i. e.,

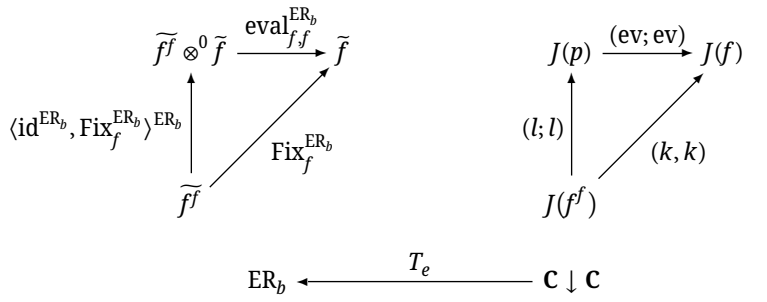
$$k = f \circ h \circ \langle \mathrm{id}, \mathrm{in}_2 \circ \Lambda(f \circ h) \rangle, \tag{5.42}$$

and hence,

$$\begin{aligned}
f \circ k \circ f^f &= f \circ (f \circ h \circ \langle \mathrm{id}, \mathrm{in}_2 \circ \Lambda(f \circ h) \rangle) \circ f^f \\
&= (f \circ h \circ \langle \mathrm{id}, \mathrm{in}_2 \circ \Lambda(f \circ h) \rangle) \circ f^f, \quad \text{by idempotent } f \\
&= k \circ f^f \quad \text{by (5.42)} \\
&= (\mathrm{fix}_b \circ \mathrm{out}_2 \circ f^f) \circ f^f \quad \text{by (5.39)} \\
&= \mathrm{fix}_b \circ \mathrm{out}_2 \circ f^f) \quad \text{by idempotent } f^f \\
&= k \quad \text{by (5.39)}
\end{aligned}$$

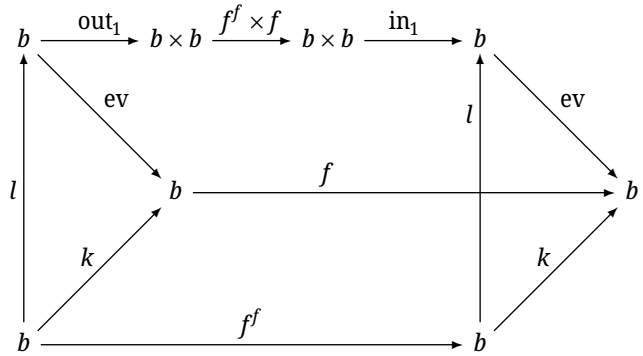
so that k is a well-defined arrow in $\mathbf{Ret}_b$, and hence, the diagram in (5.40) commutes.

What we need now is to obtain the commutative diagram for this fixed-point operator $\mathrm{Fix}_f^{\mathrm{ER}}$ in $\mathbf{ER}_b$, analog to the diagram (5.2) for the fixed-point operator fix_b of the reflexive object b in conceptually closed CCC category $\mathbf{C}$. That is, the commutative diagram of the left-hand side, as a result of application of the functor T_e to the right-hand side diagram below, by an appropriate definition of the arrow $l : b \to b$ used for the arrow $(l; l) : J(f^f) \to J(p)$ in $\mathbf{C} \downarrow \mathbf{C}$, where $p = \mathrm{in}_1 \circ (f^f \times f) \circ \mathrm{out}_1$ such that, from (5.23), $T_e^0(J(p)) = \widetilde{f^f} \otimes^0 \tilde{f}$,



(5.43)

where $\mathrm{id}^{\mathrm{ER}_b} = T_e^1(f^f; f^f)$, and the right-hand side commutative diagram in $\mathbf{C} \downarrow \mathbf{C}$ above corresponds to the following commutative diagram in $\mathbf{C}$, analog to the diagram in (5.37),



(5.44)

The upper rhombus is the commutative diagram (5.24) for $g = f$, with $\mathrm{ev} \circ p = f \circ \mathrm{ev}$, while the lower rhombus corresponds to the commutativity of the diagram in (5.40). Let us show also the commutativity of the left/right small triangles, $k = \mathrm{ev} \circ l$, by defining the arrow $l : b \to b$ based on definition (5.19) of pairing operation $\langle _, _ \rangle^{\mathrm{ER}_b}$ in ER_b, by

$$l \triangleq \mathrm{in}_1 \circ \langle f^f, k \rangle \tag{5.45}$$

where, from (5.39), $k = \mathrm{fix}_b \circ \mathrm{out}_2 \circ f^f : b \to b$. In fact, from (5.25) in this case when $g = f$, we have that $\mathrm{ev} = f \circ \mathrm{eval}_{b,b} \circ (\mathrm{out}_2 \times f) \circ \mathrm{out}_1$, and hence,

$$\begin{aligned}
\mathrm{ev} \circ l &= (f \circ \mathrm{eval}_{b,b} \circ (\mathrm{out}_2 \times f) \circ \mathrm{out}_1) \circ (\mathrm{in}_1 \circ \langle f^f, k \rangle) \\
&= f \circ \mathrm{eval}_{b,b} \circ \langle (\mathrm{out}_2 \circ f^f) \times (f \circ k) \rangle \\
&= f \circ \mathrm{eval}_{b,b} \circ \langle (\mathrm{out}_2 \circ f^f) \times k \rangle \quad \text{by (5.42) and idempotent } f, f \circ k = k \\
&= f \circ \mathrm{eval}_{b,b} \circ \langle (\mathrm{out}_2 \circ f^f) \times (\mathrm{fix}_b \circ \mathrm{out}_2 \circ f^f) \rangle \quad \text{by (5.39)} \\
&= f \circ (\mathrm{eval}_{b,b} \circ \langle \mathrm{id} \times \mathrm{fix}_b \rangle) \circ \mathrm{out}_2 \circ f^f \\
&= f \circ (\mathrm{fix}_b \circ \mathrm{out}_2 \circ f^f) \quad \text{from commutativity in (5.2)} \\
&= f \circ k = k \circ f^f \quad \text{from commutativity in (5.40)}
\end{aligned}$$

$$
\begin{aligned}
&= (\mathrm{fix}_b \circ \mathrm{out}_2 \circ f^f) \circ f^f \\
&= \mathrm{fix}_b \circ \mathrm{out}_2 \circ f^f \quad \text{from idempotent arrow } f^f \\
&= k.
\end{aligned}
$$

Thus, the complete diagram in (5.44) commutes, and hence, also the diagrams in (5.43), which define the fixed-point operators in $\mathbf{ER}_b$. This completes the proof of this theorem. □

Consequently, if we use a CCC category **C** such that is also a CoCC (conceptually-closed category), then we obtain a simplified model for untyped lambda calculus.

Corollary 12. *A CCC category* **C** *with reflexive objects, which is a conceptually closed category (CoCC) as well, with the symmetry-closure functor* $T_e : \mathbf{C} \downarrow \mathbf{C} \to \mathbf{C}$*, is a model for a type-free lambda calculus with fixed-point operators.*

It holds directly from Theorem 9 and fact that for each reflexive object b of **C**, the idempotent category $\mathbf{ER}_b$ is a strict *subcategory of* **C**.

The category $\mathbf{Ret}_b$, obtained from a CCC category **C** with a reflexive object b, has a fundamentally different internal structure from **C**, and hence, such categorial models for fixed-point operators does not permit the unification of a typed lambda calculus with fixedpoint operator extensions *into a unique* category (a model).

With this alternative way provided by Theorem 9 above, instead, by introducing the CoCC (conceptual symmetry closure) property for a CCC category **C** with reflexive objects, this category is a model both for standard typed lambda calculus, but supports also the fixed-point operators for a more general type-free language able to support the full expressive calculus power and programming with "recursive definitions."

However, the conceptual categorial symmetry permits to distinguish a program (an arrow f of **C**), viewed as such, from *the same* program viewed as data (expressed by the conceptualized object $\widetilde{f} = T_e^0(J(f))$ of **C**). That is, also in such a type-free language extension of the lambda calculus, we have the explicit difference in such a categorial framework from a program and the same program seen as data for another programs.

5.5 Topological K-theory, idempotent completion and internal categorial symmetry

In mathematics, topological K-theory is a branch of algebraic topology. It was founded to study vector bundles on topological spaces, by means of ideas now recognised as (general) K-theory that were introduced by Alexander Grothendieck in his formulation of Riemann–Rock theorem, an important theorem in mathematics, specifically in complex analysis and algebraic geometry, for the computation of the dimension of the space of meromorphic functions with prescribed zeros and allowed poles.

It relates the complex analysis of a connected compact Riemann surface with the surface's purely topological genus g (the number of "holes" of a surface, so that a sphere S^2 and a disc both have genus zero, and a torus has genus 1), in a way that can be carried over into purely algebraic settings. A Riemann surface X is a topological space that is locally homeomorphic to an open subset of $\mathbb{C}$, the set of complex numbers. In addition, the transition maps between these open subsets are required to be holomorphic. The latter condition allows one to transfer the notions and methods of complex analysis dealing with holomorphic and meromorphic functions on $\mathbb{C}$ to the surface X. For the purposes of the Riemann–Roch theorem, the surface X is always assumed to be *compact*.

The early work on topological K-theory is due to Michael Atiyah [33] and Friedrich Hirzebruch. The starting point of K-theory was a simple idea of Grothendieck [34]. If M is a semigroup, one can "symmetrize" it in the same way that we define $\mathbb{Z}$ from the semigroup $\mathbb{N}$ of natural integers. As a matter of fact, the semigroup M is difficult to compute in the applications we have in mind, whereas $S(M)$ is reachable through standard techniques of algebra and topology. Let A be a not necessarily commutative ring with unit (e. g., in topology, A is the ring of (real or complex) continuous functions on a compact space X) and let $M = \varphi(A)$ be the set of finitely generated projective (right) A-modules up to isomorphism: The group $S(M)$ obtained from M by symmetrization is called *the K-theory of* A and denoted by $K(A)$. One should notice that $K(A)$ is a "functor": a ring homomorphism $f : A \to B$ induces a group map $K(A) \to K(B)$.

Despite the apparent simplicity of this definition, K-theory has been very useful in many parts of mathematics. The category of projective modules over any ring is the *Karoubi envelope* of its full subcategory of free modules. The category of vector bundles over any paracompact space is the Karoubi envelope of its full subcategory of trivial bundles. This is in fact a special case of the previous example by the Serre–Swan theorem, and conversely this theorem can be proved by first proving both of these facts—the observation that the global sections functor is an equivalence between trivial vector bundles over X and free modules over $C(X)$ and then using the universal property of the Karoubi envelope.

Let X and Y be two topological spaces (for instance, metric spaces). Two continuous maps f_0 and f_1 from Y to X are *homotopic* (similar) if we can find a continuous map $f : Y \times [0,1] \to X$ such that $f(y,i) = f_i(y)$ for $i = 0,1$. This is an equivalence relation and we denote by $[Y,X]$ the set of equivalence classes. This set is difficult to compute explicitly in general, even for spaces as simple as spheres. Two spaces Y and X are homotopically equivalent (or have the same homotopy type) if there exist two maps $h : Y \to X$ and $g : X \to Y$ such that $k \circ g$ is homotopic to id_X and $g \circ k$ is homotopic to id_Y. The idea behind this definition is the notion of "shape" of a space, which is just another word for "homotopy type." Roughly speaking, two spaces have the same shape if they can be deformed one into another.

A variant of these considerations is to take spaces X with a "base point" x_0, and maps preserving base points. For instance, in the definition above of homotopic maps we assume moreover that $f(y_0,t) = x_0$. One reason for this variant is the definition of

the *homotopy groups* of a space X (with a fixed base point). These are defined when Y is a sphere S^n (with a fixed base point) and play an important role in algebraic topology. More precisely, the homotopy classes of base point preserving maps from the sphere S^n to X is the set of path connected components if $n = 0$, form a group if $n > 0$, which is Abelian if $n > 1$. The homotopy groups of a space are quite difficult to compute. However, as it was pointed out by Bott (Bott's theorem, Bott periodicity), the homotopy groups of Lie groups (with the obvious topology and metric induced by the natural embedding in a Euclidean space of a suitable dimension) might be easier to determine in certain circumstances.

The Brouwer fixed-point theorem was one of the early achievements of algebraic topology, and is the basis of more general fixed-point theorems, which are important in functional analysis. The theorem proved its worth in more than one way. During the 20th century, numerous fixed-point theorems were developed, and even a branch of mathematics is called fixed-point theory. Brouwer's theorem is among the foundational theorems on the topology of topological manifolds.

Brouwer's fixed-point theorem was generalized to Banach spaces. This generalization is known as Schauder's fixed-point theorem, a result generalized further by S. Kakutani to multivalued functions. The proofs of Brouwer theorems are not constructive, and Brouwer's dissatisfaction with this is partly what led him to articulate the idea of constructivity. He became the originator and zealous defender of a way of formalizing mathematics that is known as *intuitionism*.

A proof using homology uses the observation that the boundary of the n-disk D^n is S^{n-1}, the $(n-1)$-sphere. Suppose, for contradiction, that a continuous function $f : D^n \to D^n$ has no fixed point. This means that, for every point $x \in D^n$, the points x and $f(x)$ are distinct. Because they are distinct, for every point $x \in D^n$, we can construct a unique ray from $f(x)$ to x and follow the ray until it intersects the disc's boundary S^{n-1}. By calling this intersection point $F(x)$, we define a function $F : D^n \to S^{n-1}$ sending each point in the disk to its corresponding intersection point on the boundary. As a special case, whenever x itself is on the boundary, then the intersection point $F(x)$ must be x (a fixed point of F).

Consequently, F is a special type of *continuous function* known as a *retraction*: every point of the codomain (in this case S^{n-1}) is a *fixed point* of F. In effect, we have seen a particular case of the lambda calculus for which we use the reflexive objects b of a CCC category $\mathbf{C}$ with retraction $(b \Rightarrow b) < b$ via retraction pair of functions (in, out) in order to define the category of idempotents $\mathbf{Ret}_b$ and its equivalent category $\mathbf{ER}_b$ in the case when $\mathbf{C}$ is a conceptually closed category (CoCC) as well, as a particular categorial topology satisfying conceptual categorial symmetry.

In effect, Scott applied a more general Karoubi envelope in order to define this category of idempotents $\mathbf{Ret}_b$ (Definition 24 in Section 5.2) by idempotent completion of CCC. Another example of using a Karoubi envelope, which is a general idempotent completion of different kinds of categories is in the case of the exact categories [35]. In this case, the Karoubi envelope is used for the idempotent completion of the *additive categories*,

with a result that the additive category is idempotent complete iff every idempotent has a kernel: every additive category **A** can be fully faithfully embedded into an idempotent complete additive category (Remark (6.3) in [35]).

Karoubi idempotent completion for the *split* idempotents (as for the retracts) corresponds to Cauchy completion [36]. Thus, this general idempotent completion and relative embedding can be formally specified as follows.

Definition 25. The Karoubi envelope $\widehat{\mathbf{C}}$ of a category **C**, is defined by:

1. The objects of category $\widehat{\mathbf{C}}$ are the pairs (b,f) consisting of an object b of **C** and idempotent endomorphism $f : b \to b$ of **C**, such that $f \circ f = f$.
2. A morphism in $\widehat{\mathbf{C}}$ from (b,f) to (c,g) is a morphism $h : b \to c$ such that $h = g \circ h \circ f$. For any object (b,f) in $\widehat{\mathbf{C}}$, its identity arrow is $\mathrm{id}_{(b,f)} = f$.
3. The canonical embedding functor $E = (E^0, E^1) : \mathbf{C} \to \widehat{\mathbf{C}}$ is defined by

$$E^0(b) = (b, \mathrm{id}_b), \quad \text{for each object } b \text{ of } \mathbf{C}$$
$$E^1(h) = h, \quad \text{for each morphism } h \text{ in } \mathbf{C}.$$

Now we can show how Karoubi reduces (an implosion) in the case when **C** is CoCC category.

Corollary 13. *Let* **C** *be a conceptually-closed category (CoCC) with symmetry-closure functor* $T_e : \mathbf{C} \downarrow \mathbf{C} \to \mathbf{C}$. *Then its Karoubi envelope can be defined by setting each object* $(b,f) = \widetilde{f}$ *and morphism from* $h : (b,f) \to (c,g)$ *such that* $h = g \circ h \circ f$ *to morphism* $T^1_e(h;h) : \widetilde{f} \to \widetilde{g}$.

Consequently, we obtain that Karoubi envelope category is a strict subcategory of **C**, *when the canonical embedding functor E becomes the endofunctor* $T_e \circ \blacktriangle : \mathbf{C} \to \mathbf{C}$.

Proof. We only substitute each arrow $h : (b,f) \to (c,g)$ of the Karoubi envelope category (such that $h = g \circ h = h \circ f$, represented by the arrow $(h;h) : J(f) \to J(g)$ in $\mathbf{C} \downarrow \mathbf{C}$ and use the symmetry-closure functor T_e) by the arrow $T^1_e(h;h) : \widetilde{f} \to \widetilde{g}$ between idempotent conceptualized objects $\widetilde{f} = T^1_e(J(f))$ and $\widetilde{g} = T^1_e(J(g))$ in **C**.

The same arrow $h : (b,f) \to (c,g)$ is represented in $\mathbf{C} \downarrow \mathbf{C}$ also by $(h;h) : J(\mathrm{id}_b) \to J(\mathrm{id}_c)$, and hence, by $T^1_e(h;h) : \widetilde{\mathrm{id}_b} \to \widetilde{\mathrm{id}_c}$ in **C**, which is equal to $E^1(h) : E^0(b) \to E^0(c)$. Thus, the Karoubi envelope reduces to a subcategory of idempotent objects of **C**. □

With this result, for the Karoubi envelopes that generalize the theory of fixed-point operators for lambda calculus in relationship with the internal categorial symmetry theory, we finish this chapter.

6 Internal symmetry and theory of processes: Strong bisimulation of computation trees

6.1 Introduction to transition systems and their bisimulations

In order to introduce the theory of transition systems (TS) and their bisimulations, we will provide the relevant notions of this theory, which will be used in the following sections in correlation with the concept of the internal (conceptual) categorial symmetry. Transition systems are a commonly used and an understood model of computation. They provide the basic operational semantics for Milner's Calculus of Communicating Systems (CCS) and often underlie other approaches, such as that of Hoare's Communicating Sequential Processes (CSP).

Computational behaviors are often distributed, in the sense that they may be seen as spatially separated activities accomplishing a joint task. Many such systems are not meant to terminate, and hence, it makes little sense to talk about their behaviors in terms of traditional input–output functions. Rather, we are interested in the behavior of such systems in terms of the often complex patterns of stimuli/response relationships varying over time. Common to all of them is that they rest on an idea of atomic actions, over which the behavior of a system is defined.

The models differ mainly with respect to what behavioral features of systems are represented. Some models are more abstract than others, and this fact is often used in informal classifications of the models with respect to expressibility. Category theory is a convenient language for formalizing the relationships between models. This is because transition systems and their morphisms form a category, and universal constructions (including limits and colimits) of a category are determined uniquely to within isomorphism, once they exist. In fact, the universal constructions of the category of transition systems form the basis of a process description language.

In theoretical computer science, a TS is a concept used in the study of computation. It is used to describe the potential behavior of discrete systems. It consists of states and transitions between states, which may be labeled with labels chosen from a set A; the same label may appear on more than one transition. If the label set is a singleton, the system is essentially unlabeled, and a simpler definition that omits the labels is possible. In what follows, $A_\tau = A \bigcup \{\tau\}$ where $\tau \notin A$, where $\alpha, \beta, \ldots$ range over the set A and π over the set A_τ. When labels correspond to action, the τ is an internal, unobservable (*hidden*) action. A tau law is an equality that manipulates unobservable internal actions in a process P. For example, the three well-known tau laws people usually refer to were proposed by Milner [37, 38]. They are the following axioms:

$$\alpha.\tau.P = \alpha.P, \quad P + \tau.P = \tau.P, \quad \alpha.(P + \tau.Q) = \alpha.(P + \tau.Q) + \alpha.Q$$

where α is a prefix capable of inducing an α-action whereas τ is a prefix that codes up an internal action. The τ laws have been widely used in the theory of process algebra.

https://doi.org/10.1515/9783111081670-006

Processes often arise in relationship to other processes. For example, one process may refine another, or perhaps one process is a component of another. The corresponding relationships between behaviors are often expressed as morphisms between transition systems. Let us consider the following example.

Example 16 (The calculus of nondeterministic mobile processes). For more than 10 years, various calculi of mobile processes have been the focus of research in concurrency theory. These calculi are distinguished from the process calculi like CCS [38, 39] (where communication is based on the synchronization of atomic actions) in that they are capable of dealing with processes whose communication structures can change during their evolution. This ability of dynamic creation of communication link lies at the heart of mobility. In the calculi of mobile processes, mobility is achieved by passing a channel name from one process to another. The idea is simply that, instead of sending a process to another, the sender may pass to the other a name through which one can access to the process. The process that receives the channel name may communicate to a third process through that received channel name. As it turns out, the name-passing communication mechanism is a potent one.

The name-passing mechanism introduces significant difference between the algebraic theory of calculi of mobile processes and that of CCS. For example, the π-calculus of Milner, Parrow and Walker [40] is the most widely studied calculus of mobile processes. The processes are defined by BNF as follows:

$$P := \mathbf{0} \mid \pi.P \mid P + Q \mid [x = y]P$$

where a trailing $\mathbf{0}$ in prefix forms is often omitted. The process $b(x).P$ is in input prefix form. It must receive a name at b to instantiate x throughout P before P can be activated. The name x in $b(x).P$ is bound. A name appearing in P is free if it is not bound.

The process $\tau.P$ can become P after performing an internal communication. The process $[x = y]P$ behaves like either P or $\mathbf{0}$, depending on whether $x = y$ or not. The constructor $[x = y]$ is often referred to as a match operator. The choice operator "+" is well known. The process $P + Q$ acts either as P or as Q exclusively, with the choice being made nondeterministically.

The operational semantics is defined by the following labeled transition system:

$$\frac{}{\pi.P \xrightarrow{\pi} P} \quad \text{(Prefix)}$$

$$\frac{P \xrightarrow{\pi} P'}{[x = y]P \xrightarrow{\pi} P'} \quad \text{(Match)}$$

$$\frac{P \xrightarrow{\pi} P'}{P + Q \xrightarrow{\pi} P'}, \qquad \frac{Q \xrightarrow{\pi} Q'}{P + Q \xrightarrow{\pi} Q'} \quad \text{(Choice)}$$

where $\xrightarrow{\pi}$ denotes a transition obtained by the action π.

Communications in π-calculus incur changes in topological structures of the participating processes, causing the configuration of channel connections to evolve dynamically. This kind of mobility is supported by the simple mechanism of confining the contents of communications to the set of channel names.

Most people would agree that the theory of π-calculus should be far richer than that of λ-calculus. Yet we know a lot more about the λ-calculus than about the π-calculus.

Remark. Let $\Longrightarrow$ be the reflexive and transitive closure of $\xrightarrow{\tau}$.
We will write $\overset{\pi}{\Longrightarrow}$ for $\Longrightarrow \xrightarrow{\pi} \Longrightarrow$. □

Transition systems coincide mathematically with abstract rewriting systems and *directed graphs*. They differ from finite-state automata in several ways:
– The set of states is not necessarily finite, or even countable.
– The set of transitions is not necessarily finite, or even countable.

Transition systems consist of a set of states, with one initial state, together with transitions between states, which are labeled to specify the kind of events they represent.

Definition 26. A (routed) labeled transition system (LTS) is a quadruple $\mathrm{TS} = (S, i_0, A, \mathrm{Tran})$, where
– S is a set of states and $i_0 \in S$ the root (initial) state.
– A is a set of labels.
– $\mathrm{Tran} \subseteq S \times A_\tau \times S$ is the transition relation. An element of transition relation $(s_j, \pi, s_k) \in \mathrm{Tran}$ is called a transition and is usually written as $s_j \xrightarrow{\pi} s_k$.

This notation lends itself to the familiar graphical notation for transition systems.

In his foundational work on CCS (Milner, 1980), Milner introduced synchronization trees as a model of parallel processes and explained the meaning of the language of CCS in terms of operations on them. A synchronization tree is a tree together with labels on its arcs. Formally, we define synchronization trees to be special kinds of labeled transition systems, those for which every state is reachable, the transition relation is acyclic and can only branch away from the root.

There are the models where concurrency is modeled explicitly in the form of independence between actions. In some models, like Mazurkiewicz traces, the relation of independence is a basic notion while in others, like Petri nets, it is derived from something more primitive. The idea is that if two actions are enabled and also independent then they can occur concurrently, or in parallel. Models of this kind are sometimes said to capture “true concurrency,” a convenient though regrettably biased expression. They are also often called “noninterleaving models”; Petri nets and event structures can be described as forms of transition systems. Asynchronous transition systems are closely related to Petri nets. In fact, an asynchronous transition system possesses enough structure from which to construct a Petri net (its conditions are taken to be certain subsets

of transitions and states). Provided the asynchronous transition system satisfies some restrictions (entailing coherence) the constructed net will behave like the original asynchronous transition system.

Behavioral equivalences play an important role in the description of the operational semantics of concurrent systems. These equivalences are a mean for abstracting from irrelevant details introduced when describing systems as sets of states that evolve by performing actions, i. e., by means of labeled transition systems in Definition 26. Due to different opinions about what is "relevant" of concurrent systems, many different equivalences for LTS have been proposed. Many of the proposed equivalences are based on the notion of *bisimulation*. The idea of bisimulation is the most influential one in process algebra. It has been widely used in mathematical logics and game theory. The introduction of bisimulation to concurrency theory is due to the work of Park [41] and Milner [38]. Equality relations based on bisimulation are usually finer than any observational equivalence one finds useful in practice. The virtue of them is that they are subject to (co)algebraic investigation in a more or less standard manner and that it is usually easy to establish the equality of two processes by presenting a bisimulation relation.

Probably one of the most important features in process algebra is that of abstraction, since it provides us with a mechanism to hide actions that are not observable, or not interesting for any other reason. By abstraction, some of the actions in a process are made invisible or silent. Consequently, any consecutive execution of hidden steps cannot be recognized since we simply do not "see" anything happen. Many of the proposed equivalences (as, e. g., for branching and weak bisimulation) are based on the notion of bisimulation [41] that gives rise to the so-called *strong* observational equivalence over LTS. Two systems are considered as equivalent if they can perform the same labeled transitions to reach equivalent states. It leads to basic behavioral abstraction and, in the CCS languages, to elegant and simple laws [38].

It turns out that there exist many possibilities for extending bisimulation equivalence to process graphs with τ-steps. One such possible extension is incorporated in Milner"s notion of observation equivalence, also called *weak* bisimulation equivalence, which resembles ordinary bisimulation, but permits arbitrary sequences of τ-steps to precede or follow corresponding atomic actions. By modifying the basic idea of strong bisimulation, to ignore some of the labels of the transitions and some of the intermediate states, we obtain many other equivalence; for example, the *branching* observational equivalence.

Thus, formally we have the following.

Definition 27. A symmetric relation $R \subseteq S \times S$ is

- A strong bisimulation if $(s_j, s) \in R$ and $s_j \xrightarrow{\pi} s_k$ implies that there exists transition $s \xrightarrow{\pi} s'$ such that $(s_k, s') \in R$.
- A branching bisimulation if $(s, p) \in R$ and $s \Longrightarrow s_1 \xrightarrow{\pi} s_2 \Longrightarrow s_3$ implies $p \Longrightarrow p_1 \xrightarrow{\pi} p_2 \Longrightarrow p_3$ and $(s_i, p_i) \in R$ for $i = 1, 2, 3$.

Two states s and p are strongly (or branching) bisimilar if there exists a strong (or branching) bisimulation relating s and p; we will write $s \approx_s p$ or $s \approx_b p$, respectively.

While there is a general agreement about strong observational equivalence as providing the minimum of abstraction required for modeling nondeterministic systems, there is yet a discussion about its generalization. From this fact and our main consideration of a conceptual categorial symmetry and its applications, in this book we will reduce our considerations only to this basic strong observational equivalence.

6.2 Regular languages, automata and internal categorial symmetry

In this chapter, we will use algebraic methods and advocate using the category theory. To do this, we will study the *unfolding* of transition systems as trees that are obtained from the computation of a LTS while taking into account the possible choices. Consequently, by unfolding of a labeled transition system (LTS), we obtain the labeled trees and we can characterize strong, branching and weak observational equivalence as enriched functors between the categories of trees. As mentioned at the end of previous section, we will consider only the fundamental *strong* observational equivalence in Definition 27.

Labeled trees are a basic tool in both theoretical and practical computer science. Different kinds of operations and constructions are used intuitively both in programming and in the study of sequential and parallel computations. For instance, a typical feature of Milner's CCS is the manipulation of equations on infinite trees. Notice further that all the CCS-like approaches to parallel computations (for instance, Bergstra–Klop's Process Algebras [42]) lead to a similar intuition about trees, which has been also formalized in terms of category theory, for instance, by Winskel [43] and in [44].

In what follows, we will restrict our attention to *finite* trees only, i. e., to the trees for which any path (from the root to an of tree's leafs) is a finite string of actions in A, so in order to manege these paths, we need the following definition.

Definition 28. We extend the notation to strings of labels by:

1. The monoid A^* of finite sequences of actions in A with unit ϵ (action equal to the empty string). So, for each state s there exists the identity transition $s \xrightarrow{\epsilon} s$.
2. We write $s \xrightarrow{v} s_n$, where $v = a_1 a_2 \dots a_n \in A^*$ is a possibly empty string to mean

$$s \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \cdots \xrightarrow{a_n} s_n$$

   for some states $s_1, \dots, s_n$. A state s_n is said *reachable* when $i_0 \xrightarrow{v} s_n$ for some string v.
3. $\mathbf{A} = (A^*, \centerdot, \epsilon, \leq, \wedge)$ denotes the meet-semilattice monoid of strings where the operation "$\centerdot$" denotes the composition of strings, ϵ (empty string) is the bottom element of

this semilattice and $\leq$ is the prefix order[1] of words: for a given two strings $v_1, v_2 \in A^*$, the string $v = v_1 \wedge v_2$ is the largest common prefix of these two strings, and hence, $v \leq v_1$ and $v \leq v_2$.
Thus, **A** is a locally posetal 2-category.[2]

If we consider LTS as a process, which can perform sequences of elementary actions in A, then the possible future of this process is the labeled tree (the whole ramified structure of actions that it may perform) representing its *behavior*. Hence, LTSs are thought of as black boxes performing "computations": we call the computation between two processes P and Q the fragment of the behavior of P by which it becomes Q. In other words, it is the fragment after which P can perform only the actions of Q. Thus, given rooted LTS (process P represented by a labeled tree), the Q is a subtree of P. The situation has some analogies with automata theory, where behaviors and "computations" between two states of an automaton are *regular languages*, which are studied in their own right.

Thus, a locally posetal 2-category **A** of the meet-semilattice monoid $(A^*, \centerdot, \epsilon, \leq, \wedge)$ can be interpreted as the category of automata (the objects of **A**) with the partial order $\leq$ as the arrows in **A**. We can show that this category is an example of the symmetry-extended categories as follows.

Proposition 13. *The locally posetal 2-category* **A** *of the meet-semilattice monoid ordered by prefix* $(A^*, \centerdot, \epsilon, \leq, \wedge)$ *is a symmetry-extended category, conceptually closed by the symmetry-closure functor* $T_e = (T_e^0, T_e^1) : \mathbf{A} \downarrow \mathbf{A} \to \mathbf{A}$ *such that:*

1. *The object component of this functor is* $T_e^0 = J^{-1}$ *where* J^{-1}*, defined by* (1.2)*, is the inverse of the encapsulation operation* J *such that for each object* $J(v)$ *of the word* v *(an arrow in* **A***), we have that* $T_e^0(J(v)) = v$*, and hence, the conceptualized object obtained from the arrow* v *is the word* $\widetilde{v} = v$*.*
2. *The arrow component of this functor* T_e^1 *is defined by*

$$T_e^1 \triangleq \wedge(\psi F_{\mathrm{st}}^0 J; S_{\mathrm{nd}}^1). \tag{6.1}$$

---

1 Prefix order is an order (reflexive, transitive and antisymmetric) *downward total*: if $v_1 \leq v$ and $v_2 \leq v$, then $v_1 \leq v_2$ or $v_2 \leq v_1$. A partial order is defined by $v_1 \leq v_2$ iff there exists $u \in A^*$ such that $v_1 \centerdot u = v_2$ (this is the usual notion of prefix of a word).

2 As far as I know, this structure is provided by notion of bicategory [45] and its special kind of posetal bicategories (actually 2-categories).
A bicategory **B** is a set of objects $\mathrm{Ob}_{\mathbf{B}}$ together with, for each pair of objects (a, b), a category $\mathbf{B}(a, b)$, For each (a, b), the objects of category $\mathbf{B}(a, b)$ are called morphisms of **B** (or 1-cells of **B**), and they are collectively denoted by $\mathrm{Mor}_{\mathbf{B}}$. Analogously, the arrows of category $\mathbf{B}(a, b)$ are called 2-cells of **B**; for each object a of **B** is given an identity 1-cell in $\mathbf{B}(a, a)$, and for each triple of objects (a, b, c) is given a composition functor.
In particular, if $\mathbf{B}(a, b)$ is a poset, the bicategory is called locally posetal, and in this case the coherence conditions are trivially satisfied because all of the diagrams involved commute. If composition of morphisms is strictly associative and its identities are strict w. r. t. composition, the bicategories are called 2-categories [46]. Categories in their turn are special 2-categories with only identity 2-cells.

Proof. We can define the locally posetal 2-category $\mathbf{A}$ of the meet-semilattice monoid ordered by prefix $(A^*, \centerdot, \epsilon, \leq, \wedge)$, as follows:

1. The set of its objects is given by A^* of the monoid of labels with unit (empty string) ε.
2. For each pair of objects a and b, there exists a poset category $\mathbf{A}(a,b) \triangleq \{c \in A^* \mid c \leq a \wedge b\}$, whose elements are called 1-cells.
3. For each triple of objects a, b and c, the composition of the arrows (1-cells) is a monotonic operation $\mathbf{A}(a,b) \times \mathbf{A}(b,c) \longrightarrow \mathbf{A}(a,c)$, given by the meet operation $\wedge$ (largest common prefix) of the underlying semilattice monoid.
4. For each object a, its identity arrow (1-cell) id_a in $\mathbf{A}(a,a)$ is a itself (in fact, it is biggest string $c \leq a \wedge a = a$).

Thus, any arrow $c : a \to b$ between two objects (the strings in A^*) is a string $c \leq a \wedge b$, i. e., equal to a substring of the largest common prefix of the strings a and b. Consequently, both the objects and arrows in $\mathbf{A}$ are the elements in A^*, and hence $\mathbf{A}$ is a conceptually symmetric category where for each arrow c there is an object c in $\mathbf{A}$.

It is easy to show that $\mathbf{A}$ is Conceptually Closed Category (CoCC) by the covariant descending functor $T_e = (T_e^0, T_e^1) : \mathbf{A} \downarrow \mathbf{A} \to \mathbf{A}$, such that $T_e \circ \blacktriangle \simeq I_{\mathbf{A}} : \mathbf{A} \to \mathbf{A}$. In fact, let us consider the commutativity $k_1 \circ c_1 = c_2 \circ h_1$, i. e.,

$$c_1 \wedge k_1 = c_2 \wedge h_1 \tag{6.2}$$

in the higher square of the following commutative diagram in $\mathbf{A}$:

$$\begin{array}{ccc}
\begin{array}{ccc}
a_1 & \xrightarrow{c_1} & b_1 \\
{\scriptstyle h_1}\downarrow & & \downarrow{\scriptstyle k_1} \\
a_2 & \xrightarrow[c_2]{} & b_2 \\
{\scriptstyle h_2}\downarrow & & \downarrow{\scriptstyle k_2} \\
a_3 & \xrightarrow[c_3]{} & b_3
\end{array}
&
\begin{array}{c}
\\
\Leftrightarrow \\
\\
\Leftrightarrow \\
\\
\\
\end{array}
&
\begin{array}{ccc}
J(c_1) & & \widetilde{c}_1 = T_e^0(J(c_1)) = c_1 \\
\downarrow{\scriptstyle (h_1;k_1)} & & \downarrow{\scriptstyle T_e^1(h_1;k_1)} \\
J(c_2) & & \widetilde{c}_2 = T_e^0(J(c_2)) = c_2 \\
\downarrow{\scriptstyle (h_2;k_2)} & & \downarrow{\scriptstyle T_e^1(h_2;k_2)} \\
J(c_3) & & \widetilde{c}_3 = T_e^0(J(c_3)) = c_3
\end{array}
\\
\text{in } \mathbf{A} & & \mathbf{A} \downarrow \mathbf{A} \xrightarrow{T_e} \mathbf{A}
\end{array} \tag{6.3}$$

with, from point 2 above, for $i = 1,2,3$ and $j = 1,2$,

$$c_i \leq a_i \wedge b_i \quad \text{and} \quad h_j \leq a_j \wedge a_{j+1},\ k_j \leq b_j \wedge b_{j+1}. \tag{6.4}$$

Let us show that the functor T_e is well-defined also for the arrows:

– Let us show that the arrow $T_e^1(h_1;k_1) \in \mathbf{A}(c_1,c_2)$. In fact from (6.1),

$$\begin{aligned} T_e^1(h_1;k_1) &= \wedge(\psi F_{\mathrm{st}}^0 J; S_{\mathrm{nd}}^1)(h_1;k_1) \\ &= (\psi F_{\mathrm{st}}^0 J(h_1;k_1)) \wedge S_{\mathrm{nd}}^1(h_1;k_1) = \psi(J(c_1)) \wedge k_1 \end{aligned}$$

i. e.,

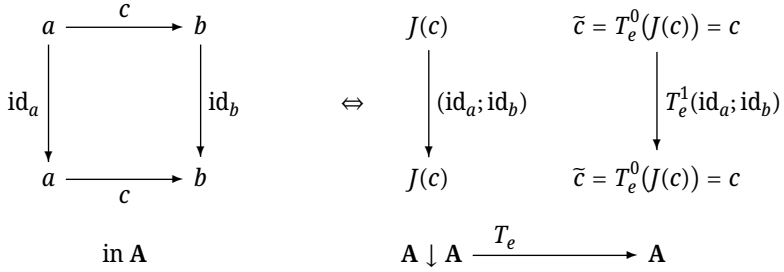
$$T_e^1(h_1;k_1) = c_1 \wedge k_1. \tag{6.5}$$

Thus,

$$\begin{aligned} T_e^1(h_1;k_1) = c_1 \wedge k_1 &= (c_1 \wedge k_1) \wedge (c_2 \wedge h_1) \quad \text{by (6.2)} \\ &= c_1 \wedge c_2 \wedge (k_1 \wedge h_1) \leq c_1 \wedge c_2, \end{aligned}$$

and hence, from point 2 above, $T_e^1(h_1;k_1) \in \mathbf{A}(c_1,c_2)$, i. e., it is an arrow $T_e^1(h_1;k_1) : c_1 \to c_2$ in $\mathbf{A}$ as presented in the diagram (6.3).

– For every identity arrow in $\mathbf{A} \downarrow \mathbf{A}$, $(\mathrm{id}_a;\mathrm{id}_b) : J(c) \to J(c)$, represented by the commutative diagram:



(6.6)

We obtain

$$\begin{aligned} T_e^1(\mathrm{id}_a;\mathrm{id}_b) &= \wedge(\psi F_{\mathrm{st}}^0 J; S_{\mathrm{nd}}^1)(\mathrm{id}_a;\mathrm{id}_b) \\ &= (\psi F_{\mathrm{st}}^0 J(\mathrm{id}_a;\mathrm{id}_b)) \wedge S_{\mathrm{nd}}^1(\mathrm{id}_a;\mathrm{id}_b) \\ &= \psi(J(c)) \wedge \mathrm{id}_b = c \wedge \mathrm{id}_b \\ &= c \wedge b \quad \text{by point 4 above} \\ &= c, \end{aligned}$$

from the fact that arrow (1-cell) $c \in \mathbf{A}(a,b)$, and hence, $c \leq a \wedge b \leq b$.
Thus, from the fact that, from point 4 above, $\mathrm{id}_c = c$, we obtain that $T_e^1(\mathrm{id}_a;\mathrm{id}_b) = \mathrm{id}_c$ is the identity arrow in $\mathbf{A}$.

– For a composition of arrows, from diagram in (6.3) and from (6.5), we have that

$$\begin{aligned}
T_e^1(h_2;k_2)\circ T_e^1(h_1;k_1) &= (c_2\wedge k_2)\circ(c_1\wedge k_1)\\
&= (c_2\wedge k_2)\wedge(c_2\wedge h_1)\text{by (6.2)}\\
&= k_2\wedge c_2\wedge h_1\\
&= k_2\circ(c_2\circ h_1)\\
&= k_2\circ(k_1\circ c_1)\quad\text{by (6.2)}\\
&= k_2\wedge(k_1\wedge c_1) = c_1\wedge(k_2\wedge k_1) = c_1\wedge(k_2\circ k_1)\\
&= c_1\wedge S^1_{\mathrm{nd}}(h_2\circ h_1;k_2\circ k_1)\\
&= \psi F^0_{\mathrm{st}}J(h_2\circ h_1;k_2\circ k_1)\wedge S^1_{\mathrm{nd}}(h_2\circ h_1;k_2\circ k_1)\\
&= \wedge(\psi F^0_{\mathrm{st}}J;S^1_{\mathrm{nd}})(h_2\circ h_1;k_2\circ k_1)\\
&= T_e^1(h_2\circ h_1;k_2\circ k_1).
\end{aligned}$$

For every object a, we have that

$$\begin{aligned}
T_e^0\blacktriangle^0(a) &= T_e^0(J(\mathrm{id}_a))\\
&= T_e^0(J(a))\quad\text{by point 4 above}\\
&= a = I_{\mathbf{A}}(a)
\end{aligned}$$

and hence, it is valid the natural isomorphism $\varphi : T_e\blacktriangle \simeq I_{\mathbf{A}}$ with $\varphi(a) = a = \varphi^{-1}(a)$. Let us demonstrate that $\mathbf{A}$ is a symmetry-extended category, i. e., that $\tau^{-1}\bullet\tau = \psi$. In fact, for each arrow $a_1 : a\to b$ in $\mathbf{A}$, and hence, $(a_1;a_1) : J(\mathrm{id}_a)\to J(\mathrm{id}_b)$ in $\mathbf{A}\downarrow\mathbf{A}$, from (3.16),

$$\begin{aligned}
\tau^{-1}\bullet\tau(J(a_1)) &= ((\varphi S_{\mathrm{nd}})\bullet(T_e(\psi;\psi))\bullet(\varphi^{-1}F_{\mathrm{st}}))(J(a_1))\\
&= \varphi(b)\circ T_e(a_1;a_1)\circ\varphi^{-1}(a)\\
&= b\circ(\wedge(\psi F^0_{\mathrm{st}}J;S^1_{\mathrm{nd}})(a_1;a_1))\circ a\\
&= b\circ(\mathrm{id}_a\wedge a_1)\circ a = b\circ(\mathrm{id}_a\circ a_1)\circ a\\
&= b\circ a_1\circ a = b\wedge a_1\wedge a\\
&= a_1\quad\text{by } a_1\le a\wedge b\text{, from point 2 above for 1-cell } a_1 : a\to b\\
&= \psi(J(a_1)).
\end{aligned}$$

□

The trees in which arcs are labeled (finite labeled trees) are a model, which underpin much of the work in the semantics of parallel computation. The nodes are thought as states and arcs as transitions (events) with branching representing nondeterminism.

Definition 29 ([43, 47]). A tree t is a subset of finite sequence in A^* of transitions (events) called computations, closed under initial subsequences. Empty string ϵ is a tree composed by only root node. We define

$$t_1 \rightarrow_T t_2 \quad \text{iff} \quad \exists a \in A \quad \text{such that } t_2 = t_1 a$$

i. e., t_2 is a concatenation of the sequences t_1 and a.

A morphism of trees from t_1 to t_2 is a map $f : t_1 \to t_2$ such that $f(\epsilon) = \epsilon$ and

$$s \rightarrow_{t_1} s' \quad \text{implies} \quad f(s) = f(s') \quad \text{or} \quad f(s) \rightarrow_{t_2} f(s') \tag{6.7}$$

so that it preserves the root-node and either preserves or collapses arcs.

Here, we consider this definition of trees that have been also used to model concurrent systems [48] and nondeterministic regular expressions. However, we will not use their definition of the category of trees but a new more general version of the morphisms w. r. t. simple function (6.7), which will be introduced in the next section.

A single tree is modeled also by specifying its *paths*, based on the computations (defined as objects of the posetal 2-category **A**) along each of them (the *extent*), and the part of the extent in which pairs of computations agree (the *agreement*): this agreement is a formal definition of the property (in Definition 29) that the paths of the tree are "closed under initial subsequences." So, the path labels are elements of a free monoid A^*, for some finite alphabet A.

Definition 30 (Finite labeled trees [48]). A finite, monoid-labeled by $(A^*, \centerdot, \epsilon, \leq, \wedge)$, tree $t = (X, \text{ext}, \text{agr})$ is a set X of paths, equipped by two functions

- $\text{ext} : X \to A^*$, the *extent map*, giving the computation $\text{ext}(x)$ along a path $x \in X$,
- $\text{agr} : X \times X \to A^*$, establishing the *agreement* $\text{agr}(x, y)$ between pairs of computations $\text{ext}(x)$ and $\text{ext}(y)$,

such that for each $x, y, z \in X$, we require that:

1. $\text{agr}(x, x) = \text{ext}(x)$
2. $\text{agr}(x, y) = \text{agr}(y, x)$
3. $\text{agr}(x, y) \leq \text{ext}(x) \wedge \text{ext}(y)$
4. $\text{agr}(x, y) \wedge \text{agr}(y, z) \leq \text{agr}(x, z)$

These amount to requiring that a path agrees with itself along all its length; agreement is symmetric; the agreement between two paths is not bigger than their largest common prefix (paths are forced to agree on a common initial segment and they cannot join up again once split); the common agreement between paths x, y and z is not bigger than that between x and z.

Based on the definition above for the finite labeled trees, we can use the notion of unfolding of a transition system that allows us to associate a labeled tree to each labeled transition system.

6.3 Symmetry-extended category of finite labeled trees

The categories of trees are generalized toward the categories of transition systems. The pleasant categorical set-up, which exists between the categories of trees and transition systems makes possible a smooth translation between *operational semantics* expressed in terms of transition systems and *denotational semantics* expressed in terms of trees. For example, synchronization trees [43] are a basic, very concrete, interleaving model of *parallel computation* in which processes communicate by mutual synchronization. A synchronization tree is a tree in which the nodes represent states and the arcs represent event (transition) occurrences, labeled to show how they synchronize with events in the environment. Synchronization trees are a basic, very concrete, interleaving model of parallel computation in which processes communicate by mutual synchronization. A synchronization tree is a tree in which the nodes represent states and the arcs represent event occurrences, labeled to show how they synchronize with events in the environment.

Based on the Definition 30 of finite labeled trees (FLT), we will introduce a new category of finite labeled trees such that both its objects and morphisms (arrows) are the FLTs, but the morphism between the objects (trees) satisfy also the mapping requirement (6.7) in the original Definition 29.

Definition 31 (Category of finite Labeled Trees). Let $\mathbf{Z}$ denote the category of finite labeled trees composed by:

1. The objects are the trees $(X_i, \mathrm{ext}_i, \mathrm{agr}_i)$ composed by the set X_i of paths, extent map $\mathrm{ext}_i : X_i \to A^*$ and agreement $\mathrm{agr}_i : X_i \times X_i \to A^*$ satisfying the four requirements of Definition 30.
2. Given two objects (trees) $t_1 = (X_1, \mathrm{ext}_1, \mathrm{agr}_1)$ and $t_2 = (X_2, \mathrm{ext}_2, \mathrm{agr}_2)$, then there exist an arrow $t_{12} : t_1 \to t_2$ in $\mathbf{Z}$, which is a tree $(X_1, \mathrm{ext}_1, \mathrm{agr}_{12})$, if the following requirements are satisfied:
   - The set of computations of t_1 is a subset of computations of t_2, i. e., $\{\mathrm{ext}_1(x_i) \mid x_i \in X_1\} \subseteq \{\mathrm{ext}_2(y_i) \mid y_i \in X_2\}$
   - There exists a map $f_{12} : X_1 \to X_2$ such that $\mathrm{ext}_2(f_{12}(x_k)) = \mathrm{ext}_1(x_k)$ for any $x_k \in X_1$, and for any two paths $x_i, x_j \in X_1$,

$$\mathrm{agr}_{12}(x_i, x_j) \triangleq \mathrm{agr}_2(f_{12}(x_i), f_{12}(x_j)) \geq \mathrm{agr}_1(x_i, x_j). \tag{6.8}$$

   So, for each object (tree) $t_1 = (X_1, \mathrm{ext}_1, \mathrm{agr}_1)$, its identity arrow id_{t_1} (with identity map $f_1 = \mathrm{id}_{X_1} : X_1 \to X_1$) is the tree t_1 itself.
3. Let $t_3 = (X_3, \mathrm{ext}_3, \mathrm{agr}_3)$ be the codomain object of an arrow $t_{23} : t_2 \to t_3$ with its map $f_{23} : X_2 \to X_3$, so that $t_{23} = (X_2, \mathrm{ext}_2, \mathrm{agr}_{23})$. The *composition* of the arrows $t_{13} = t_{23} \circ t_{12} : t_1 \to t_3$ with its map $f_{13} = f_{23} \circ f_{12} : X_1 \to X_3$ is defined by the tree

$$t_{23} \circ t_{12} = (X_2, \mathrm{ext}_2, \mathrm{agr}_{23}) \circ (X_1, \mathrm{ext}_1, \mathrm{agr}_{12}) \triangleq (X_1, \mathrm{ext}_1, \mathrm{agr}_{13}) \tag{6.9}$$

where, from (6.8), for any $x_i, x_j \in X_1$,

$$\mathrm{agr}_{13}(x_i, x_j) = \mathrm{agr}_3(f_{13}(x_i), f_{13}(x_j)) = \mathrm{agr}_{23}(f_{12}(x_i), f_{12}(x_j)). \tag{6.10}$$

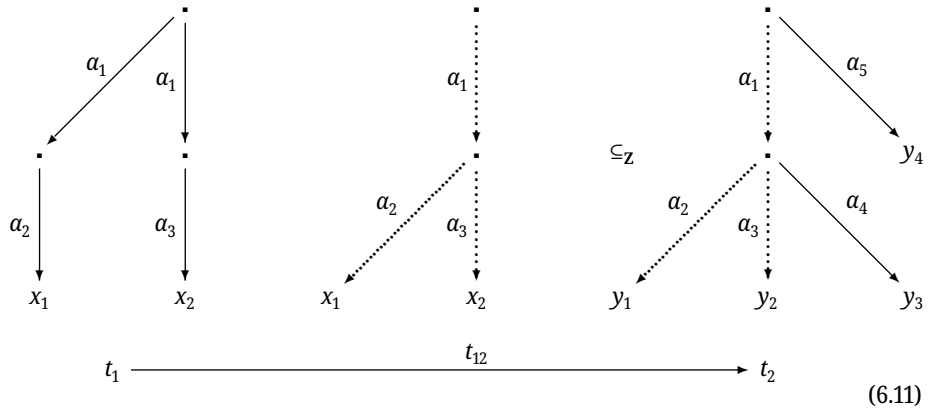
So, for any $x_i \in X_1$,

$$\mathrm{ext}_3(f_{13}(x_i)) = \mathrm{ext}_2(f_{12}(x_i)) = \mathrm{ext}_1(x_i) \quad \text{and}$$
$$\{\mathrm{ext}_1(x_i) \mid x_i \in X_1\} \subseteq \{\mathrm{ext}_2(y_i) \mid y_i \in X_2\} \subseteq \{\mathrm{ext}_3(z_i) \mid z_i \in X_3\}.$$

It is easy to show that definition of composition of the arrows is well-defined, i. e., for the composed arrow $t_{13} = t_{23} \circ t_{12} : (X_1, \mathrm{ext}_1, \mathrm{agr}_1) \to (X_3, \mathrm{ext}_3, \mathrm{agr}_3)$, with $t_{13} = (X_1, \mathrm{ext}_1, \mathrm{agr}_{13})$, the requirement (6.8) is satisfied, i. e., for each $x_i, x_j \in X_1$, $\mathrm{agr}_{13}(x_i, x_j) = \mathrm{agr}_{23}(f_{12}(x_i), f_{12}(x_j)) = \mathrm{agr}_{12}(x_i, x_j) \geq \mathrm{agr}_1(x_i, x_j)$.

For each identity arrow $\mathrm{id}_t : t \to t$, where $t = (X, \mathrm{ext}, \mathrm{agr})$, from definition of the arrows in **Z** it holds that id_t is the tree $\mathrm{id}_t = (X, \mathrm{ext}, \mathrm{agr}')$ such that for each two $x_i, x_j \in X$, $\mathrm{agr}'(x_i, x_j) = \mathrm{agr}(x_i, x_j)$, and hence, $\mathrm{id}_t = t$.

Example 17. Let us consider the objects (trees) $t_k = (X_k, \mathrm{ext}_k, \mathrm{agr}_k)$, for $k = 1, 2$, where $X_1 = \{x_1, x_2\}$ and $X_2 = \{y_1, y_2, y_3, y_4\}$, with the arrow $t_{12} : t_1 \to t_2$ (a tree $t_{12} = (X_1, \mathrm{ext}_1, \mathrm{agr}_{12})$) in the category **Z** where $f_{12} : X_1 \to X_2$ such that $f_{12}(x_1) = y_1$ and $f_{12}(x_2) = y_2$ (note that the arrow tree is represented by dotted transitions),



(6.11)

with $\mathrm{ext}_1(x_1) = \mathrm{ext}_2(f_{12}(x_1)) = \alpha_1\alpha_2$, $\mathrm{ext}_1(x_2) = \mathrm{ext}_2(f_{12}(x_2)) = \alpha_1\alpha_3$, $\mathrm{agr}_1(x_1, x_2) = \epsilon \leq \alpha_1 = \mathrm{agr}_{12}(x_1, x_2) = \mathrm{agr}_2(f_{12}(x_1), f_{12}(x_2)) = \mathrm{agr}_2(y_1, y_2)$.

We can introduce the subtree relation $\subseteq_{\mathbf{Z}}$, such that for given two trees $t_k = (X_k, \mathrm{ext}_k, \mathrm{agr}_k)$ and $t_n = (X_n, \mathrm{ext}_n, \mathrm{agr}_n)$ with the path's map $f_{kn} : X_k \to X_n$,

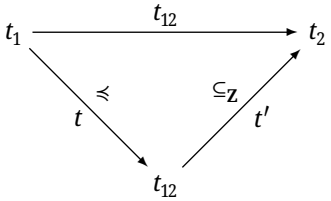
$$t_k \subseteq_{\mathbf{Z}} t_n \quad \text{iff} \quad \{\mathrm{ext}_k(x_i) \mid x_i \in X_k\} \subseteq \{\mathrm{ext}_n(y_i) \mid y_i \in X_n\} \quad \text{and}$$
$$\forall x_i, x_j \in X_k.(\mathrm{agr}_k(x_i, x_j) = \mathrm{agr}_n(f_{kn}(x_i), f_{kn}(x_j))). \tag{6.12}$$

Consequently, for the arrow $t_{12} : t_1 \to t_2$ above, we have that the tree-arrow t_{12} is a subtree of the codomain object t_2 of this arrow, i. e.,

$$t_{12} \subseteq_{\mathbf{Z}} t_2$$

and this fact is graphically represented by the dotted transitions in the diagram above.

Note that this subtree relation can be represented by the arrow $t' = (X_1, \mathrm{ext}_1, \mathrm{agr}_{12}) : t_{12} \to t_2$ in the category $\mathbf{Z}$ and we have also the arrow $t = (X_1, \mathrm{ext}_1, \mathrm{agr}_{12}) : t_1 \to t_{12}$, so that the composition of these two derived arrows $t' \circ t$ is equal to the original arrow $t_{12} : t_1 \to t_2$. That is, each arrow $t_{12} : t_1 \to t_2$ can be decomposed as a following composition of two, derived from it, arrows:



(6.13)

Note that t and t' are equal to $t_{12} = (X_1, \mathrm{ext}_1, \mathrm{agr}_{12})$.

This fact, as will be shown in what follows, is an indication of a particular conceptual symmetry of this category of trees $\mathbf{Z}$.

Moreover, we can define a partial order $\preccurlyeq$ between the trees $t_k = (X_k, \mathrm{ext}_k, \mathrm{agr}_k)$ and $t_n = (X_n, \mathrm{ext}_n, \mathrm{agr}_n)$, and we have the same extensions of their paths, i. e., $X_n = X_k$ and $\mathrm{ext}_n = \mathrm{ext}_k$, by

$$t_k \preccurlyeq t_n \quad \text{iff} \quad \forall x_i, x_j \in X_k.(\mathrm{agr}_k(x_i, x_j) \leq \mathrm{agr}_n(x_i, x_j)) \tag{6.14}$$

so that, for each arrow $t_{12} : t_1 \to t_2$ from (6.14) and (6.8), we obtain the following relationship between the trees:

$$t_1 \preccurlyeq t_{12} \subseteq_{\mathbf{Z}} t_2$$

as indicated by composition of the arrow $t : t_1 \xrightarrow{\preccurlyeq} t_{12}$ and the arrow $t' : t_{12} \xrightarrow{\subseteq_{\mathbf{Z}}} t_2$ in the diagram (6.13).

Well-known operations on trees arise as categorial constructions. For example, a sum construction is a *coproduct* on trees while many familiar parallel compositions of synchronization trees are restrictions of the *product* in the underlying category of trees. Thus, as in other previously proposed tree categories (see, e. g., categorial constructions in [48]), we have the following categorial structures in the category $\mathbf{Z}$ of trees.

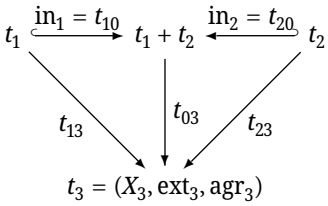
Proposition 14. *The category* $\mathbf{Z}$ *has the coproducts, such that for each two arrows* $t_k = (X_k, \mathrm{ext}_k, \mathrm{agr}_k)$ *for* $k = 1, 2$ *the coproduct object* $t_0 = t_1 + t_2$ *is obtained by their gluing by the roots,* $t_0 = (X_1 \biguplus X_2, \mathrm{ext}_1 + \mathrm{ext}_2, \mathrm{agr}_1 + \mathrm{agr}_2)$, *where:*

1. *Disjoint union of paths,* $X_1 \biguplus X_2 = S_1 \bigcup S_2$*, where* $S_k \triangleq \{(k, x_i) \mid x_i \in X_k\}$ *for* $k = 1, 2$.
2. *Extension function* $(\mathrm{ext}_1 + \mathrm{ext}_2) : X_1 \biguplus X_2 \to A^*$*, such that* $(\mathrm{ext}_1 + \mathrm{ext}_2)(1, x_i) = \mathrm{ext}_1(x_i)$ *and* $(\mathrm{ext}_1 + \mathrm{ext}_2)(2, x_i) = \mathrm{ext}_2(x_i)$.
3. *Agreement,* $(\mathrm{agr}_1 + \mathrm{agr}_2) : (X_1 \biguplus X_2) \times (X_1 \biguplus X_2) \to A^*$*, such that for* $k, n \in \{1, 2\}$, $(\mathrm{agr}_1 + \mathrm{agr}_2)((k, x_i), (n, x_j))$ *is equal to*
   (a) *empty string* ϵ*, if* $k \neq n$;
   (b) *agreement* $\mathrm{agr}_k(x_i, x_j)$*, otherwise,*

with two monic arrows into this coproduct object, for $k = 1, 2$, $\mathrm{in}_k = t_{k0} : t_k \hookrightarrow t_0$*, such that* $t_{k0} = t_k$ *and path's mappings* $f_{k0} : X_k \to X_1 \biguplus X_2$*, such that for each* $x_i \in X_k$*,*

$$f_{k0}(x_i) = (k, x_i) \tag{6.15}$$

Proof. Let us consider the commutative diagram for this product object $t_0 = t_1 + t_2$,



with two arrows, for $k = 1, 2$, $t_{k3} = (X_k, \mathrm{ext}_k, \mathrm{agr}_{k3})$ with the path's maps $f_{k3} : X_k \to X_3$, and we need to show that, for this commutative diagram, the arrow $t_{0,3} = (X_1 \biguplus X_2, \mathrm{ext}_1 + \mathrm{ext}_2, \mathrm{agr}_{0,3})$, with the path's map $f_{03} : X_1 \biguplus X_2 \to X_3$, is unique.

From the commutativity, it must be satisfied that

$$f_{13} = f_{03} \circ f_{10} \quad \text{and} \quad f_{23} = f_{03} \circ f_{20} \tag{6.16}$$

so that from the definition of t_{03} we need to show only that its agreement function agr_{03} is unique for given two arrows t_{13} and t_{23}.

In fact, for any $x_i, x_j \in X_k$, for $k = 1, 2$, from the commutativity it must be satisfied that

$$\mathrm{agr}_{03}(f_{k0}(x_i), f_{k0}(x_j)) = \mathrm{agr}_3(f_{13}(x_i), f_{13}(x_j)) = \mathrm{agr}_{k3}(x_i, x_j)$$

and hence, agr_{03} is uniquely defined from the agreements agr_{13} and agr_{23} of the two arrows t_{13} and t_{23}, respectively. □

So, based on the Example 17, we have the following categorial property of the category of trees **Z**.

Proposition 15. *There exist the following two fundamental properties of the category of trees* **Z***:*

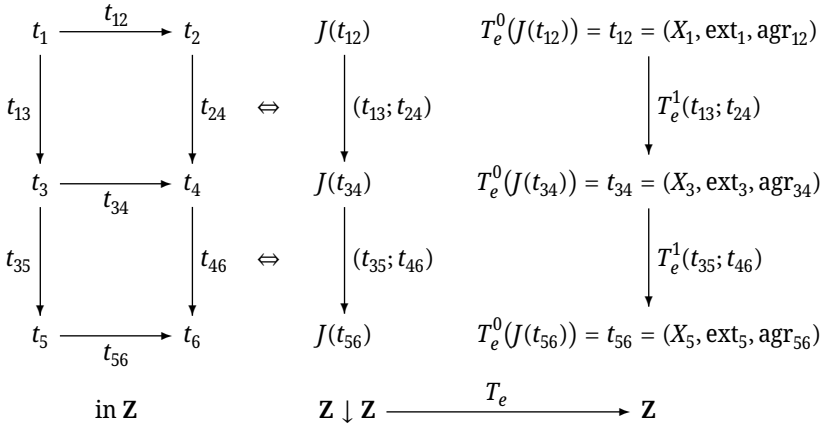
1. *It is a locally posetal 2-category.*
2. *It is a symmetry-extended category, closed by the symmetry-closure functor* $T_e = (T_e^0, T_e^1) : \mathbf{Z} \downarrow \mathbf{Z} \to \mathbf{Z}$*, such that its object component* $T_e^0 = \psi = J^{-1}$ *while the arrow component* T_e^1 *is defined by the following triple of functions:*

$$T_e^1 \triangleq (p_1\psi\,\mathrm{dom}, p_2\psi\,\mathrm{dom}, \cdot(p_3\psi\,\mathrm{cod}, [F_{\mathrm{st}}^1])) \tag{6.17}$$

where dom, cod *returns by domain and codomain objects of the arrows in* $\mathbf{Z} \downarrow \mathbf{Z}$*, while* p_i*, for* $i = 1, 2, 3$ *returns with i-th component of tree triples* $(X, \mathrm{ext}, \mathrm{agr})$ *and for each arrow* $t_{kn} : t_k \to t_n$ *the function "*$[_]$*" returns with the product of the path's map of* t_{kn}*, i. e.,* $[t_{kn}] = f_{kn} \times f_{kn} : X_k \times X_k \to X_n \times X_n$*.*

Proof. Point 1: for each two objects (trees) $t_k = (X_k, \mathrm{ext}_k, \mathrm{agr}_k)$ and $t_n = (X_n, \mathrm{ext}_n, \mathrm{agr}_n)$ for the category $\mathbf{Z}(t_k, t_n)$, its objects are 1-cells, $t_{kn} = (X_k, \mathrm{ext}_k, \mathrm{agr}_{kn})$, $t'_{kn} = (X_k, \mathrm{ext}_k, \mathrm{agr}'_{kn})$, which are the trees with the same paths and their extensions but possibly different agreements, so that the arrows between them are partial orders $t : t_{kn} \xrightarrow{\preccurlyeq} t'_{kn}$, as defined by the p. o. relation $\preccurlyeq$ in (6.14). Thus, each $\mathbf{Z}(t_k, t_n)$ is a poset category, for each object t its 1-cell identity id_t in $\mathbf{Z}(t, t)$ is t itself, so that $\mathbf{Z}$ is a locally posetal 2-category.

Point 2: Let $t_{24} \circ t_{12} = t_{34} \circ t_{13}$ be a commutative diagram in $\mathbf{Z}$, of the arrows $t_{jk} : t_j \to t_k$ (with path's maps $f_{jk} : X_j \to X_k$) between the objects (trees) $t_n = (X_n, \mathrm{ext}_n, \mathrm{agr}_n)$ for $n = 1, 2, 3, 4$, corresponding to the arrow $(t_{13}; t_{24}) : J(t_{12}) \to J(t_{34})$, and hence,



(6.18)

with the arrow

$$\begin{aligned} T_e^1(t_{13}, t_{24}) &\triangleq (p_1\psi\,\mathrm{dom}, p_2\psi\,\mathrm{dom}, \cdot(p_3\psi\,\mathrm{cod}, [F_{\mathrm{st}}^1]))(t_{13}, t_{24}) \\ &= (p_1(t_{12}), p_2(t_{12}), \cdot(p_3(t_{34}), [t_{13}])) \\ &= (p_1(t_{12}), p_2(t_{12}), p_3(t_{34}) \cdot (f_{13} \times f_{13})) \\ &= (p_1(X_1, \mathrm{ext}_1, \mathrm{agr}_{12}), p_2(X_1, \mathrm{ext}_1, \mathrm{agr}_{12}), p_3(X_3, \mathrm{ext}_3, \mathrm{agr}_{34}) \cdot (f_{13} \times f_{13})) \end{aligned}$$

$$= (X_1, \mathrm{ext}_1, \mathrm{agr}_{34} \cdot (f_{13} \times f_{13}))$$

with the path's map $f_{13} : X_1 \to X_3$, so that for each $x_i, x_j \in X_1$, $\mathrm{agr}_{34} \cdot (f_{13} \times f_{13})(x_i, x_j) = \mathrm{agr}_{34}(f_{13}(x_i), f_{13}(x_j))$. Thus, this arrow in $\mathbf{Z}$ is well-defined if

$$\text{for each } x_i, x_j \in X_1.(\mathrm{agr}_{12}(x_i, x_j) \leq \mathrm{agr}_{34}(f_{13}(x_i), f_{13}(x_j))).$$

In fact, from the commutative diagram above, $t_{24} \circ t_{12} = t_{34} \circ t_{13} : (X_1, \mathrm{ext}_1, \mathrm{agr}_1) \to (X_4, \mathrm{ext}_4, \mathrm{agr}_4)$, we obtain that for each $x_i, x_j \in X_1$,

$$\begin{aligned}\mathrm{agr}_{34}(f_{13}(x_i), f_{13}(x_j)) &= \mathrm{agr}_4(f_{34}f_{13}(x_i), f_{34}f_{13}(x_j)) = \mathrm{agr}_4(f_{24}f_{12}(x_i), f_{24}f_{12}(x_j)) \\ &= \mathrm{agr}_{24}(f_{12}(x_i), f_{12}(x_j)) \\ &\geq \mathrm{agr}_{12}(x_i, x_j)\end{aligned}$$

so that the arrow $T_e^1(t_{13}, t_{24}) : t_{12} \to t_{34}$ is well-defined as shown in scheme (6.18).

For the functorial property on the composition of the arrows, we have

$$\begin{aligned}T_e^1(t_{35}, t_{46}) \circ T_e^1(t_{13}, t_{24}) &= (X_3, \mathrm{ext}_3, \mathrm{agr}_{56} \cdot (f_{35} \times f_{35})) \circ (X_1, \mathrm{ext}_1, \mathrm{agr}_{34} \cdot (f_{13} \times f_{13})) \\ &= (X_1, \mathrm{ext}_1, \mathrm{agr}_{56} \cdot ((f_{35} \circ f_{13}) \times (f_{35} \circ f_{13}))) \\ &\quad \text{from composition (6.8) and (6.10)} \\ &= (X_1, \mathrm{ext}_1, \mathrm{agr}_{56} \cdot (f_{15} \times f_{15})) \quad \text{from } f_{15} = f_{35} \circ f_{13} \\ &= T_e^1(t_{35} \circ t_{13}; t_{46} \circ t_{24})\end{aligned}$$

For every object (a tree) t, we have that

$$\begin{aligned}T_e^0 \blacktriangle^0(t) &= T_e^0(J(\mathrm{id}_t)) \\ &= T_e^0(J(t)) \quad \text{by point 2 of Definition 31} \\ &= t = I_{\mathbf{Z}}(t)\end{aligned}$$

and hence, the natural isomorphism $\varphi : T_e \blacktriangle \simeq I_{\mathbf{Z}}$ is valid with, for any object (a tree) t,

$$\varphi(t) = t = \varphi^{-1}(t). \tag{6.19}$$

Let us demonstrate that $\mathbf{Z}$ is a symmetry-extended category, i. e., that $\tau^{-1} \bullet \tau = \psi$. Let us consider, for two objects $t_k = (X_k, \mathrm{ext}_k, \mathrm{agr}_k)$ for $k = 1, 2$, an arrow $t_{12} = (X_1, \mathrm{ext}_1, \mathrm{agr}_{12}) : t_1 \to t_2$ with a path's map $f_{12} : X_1 \to X_2$, so that from (1.13), $\sigma(J(t_{12})) = (\psi \blacktriangle^0 F^0_{\mathrm{st}}; \psi)(J(t_{12})) = (\psi(J(\mathrm{id}_{t_1})); t_{12})$, and $\sigma^{-1}(J(t_{12})) = (\psi; \psi \blacktriangle^0 S^0_{\mathrm{nd}})(J(t_{12})) = (t_{12}; \psi(J(\mathrm{id}_{t_2})))$, i. e.,

$$\sigma(J(t_{12})) = (\mathrm{id}_{t_1}; t_{12}) \quad \text{and} \quad \sigma^{-1}(J(t_{12})) = (t_{12}; \mathrm{id}_{t_2}). \tag{6.20}$$

Thus, from (3.15), we obtain

$$\begin{aligned}\tau(J(t_{12})) &= (T_e^1\sigma) \bullet (\varphi^{-1}F_{\mathrm{st}}^0)(J(t_{12}))\\ &= T_e^1\sigma(J(t_{12})) \circ \varphi^{-1}(t_1)\\ &= T_e^1(\mathrm{id}_{t_1}; t_{12}) \circ t_1 \quad \text{by (6.19) and (6.20)}\\ &= T_e^1(\mathrm{id}_{t_1}; t_{12}) \circ \mathrm{id}_{t_1} \quad \text{by the identity arrow } \mathrm{id}_{t_1} = t_1 : t_1 \to t_1\end{aligned}$$

and

$$\begin{aligned}\tau^{-1}(J(t_{12})) &= (\varphi S_{\mathrm{nd}}) \bullet (T_e\sigma^{-1})(J(t_{12}))\\ &= \varphi^{-1}(t_2) \circ T_e^1\sigma^{-1}(J(t_{12}))\\ &= t_2 \circ T_e^1(t_{12}; \mathrm{id}_{t_2}) \quad \text{by (6.19) and (6.20)}\\ &= \mathrm{id}_{t_2} \circ T_e^1(t_{12}; \mathrm{id}_{t_2}) \quad \text{by the identity arrow } \mathrm{id}_{t_2} = t_2 : t_2 \to t_2.\end{aligned}$$

That is, we obtain the following arrows:

$$\tau(J(t_{12})) = T_e^1(\mathrm{id}_{t_1}; t_{12}) \circ \mathrm{id}_{t_1} : t_1 \to t_{12} \quad \text{and} \quad \tau^{-1}(J(t_{12})) = t_2 \circ T_e^1(t_{12}; \mathrm{id}_{t_2}) : t_{12} \to t_2 \tag{6.21}$$

where from the commutative diagram

$$\begin{array}{ccc} t_1 & \xrightarrow{t_1 = \mathrm{id}_{t_1}} & t_1 \\ {\scriptstyle \mathrm{id}_{t_1}}\downarrow & & \downarrow{\scriptstyle t_{12}} \\ t_1 & \xrightarrow[t_{12}]{} & t_2 \end{array} \quad \Leftrightarrow \quad \begin{array}{ccc} J(t_1) & & t_1 = T_e^0(J(t_1)) \\ \downarrow{\scriptstyle (\mathrm{id}_{t_1}; t_{12})} & & \downarrow{\scriptstyle T_e^1(\mathrm{id}_{t_1}; t_{12})} \\ J(t_{12}) & & t_{12} = T_e^0(J(t_{12})) \end{array}$$

$$\text{in } \mathbf{Z} \qquad\qquad \mathbf{Z} \downarrow \mathbf{Z} \xrightarrow{T_e} \mathbf{Z}$$

and from the fact that to the identity arrow $\mathrm{id}_{t_1} : t_1 \to t_1$ is associated the identity path's map $f_{11} = \mathrm{id}_{X_1} : X_1 \to X_1$, and from the definition of T_e^1, we obtain

$$\begin{aligned}&T_e^1(\mathrm{id}_{t_1}; t_{12}) = (X_1, \mathrm{ext}_1, \mathrm{agr}_{12} \cdot (f_{11} \times f_{11})) = (X_1, \mathrm{ext}_1, \mathrm{agr}_{12}), \quad \text{and from (6.21),}\\ &\tau(J(t_{12})) = T_e^1(\mathrm{id}_{t_1}; t_{12}) = t_{12}\end{aligned} \tag{6.22}$$

and from the commutative diagram

$$\begin{array}{ccc} t_1 & \xrightarrow{t_{12}} & t_2 \\ {\scriptstyle t_{12}}\downarrow & & \downarrow{\scriptstyle \mathrm{id}_{t_2}} \\ t_2 & \xrightarrow[t_2]{} & t_2 \end{array} \quad \Leftrightarrow \quad \begin{array}{ccc} J(t_{12}) & & t_{12} = T_e^0(J(t_{12})) \\ \downarrow{\scriptstyle (t_{12}; \mathrm{id}_{t_2})} & & \downarrow{\scriptstyle T_e^1(t_{12}; \mathrm{id}_{t_2})} \\ J(t_2) & & t_2 = T_e^0(J(t_2)) \end{array}$$

$$\text{in } \mathbf{Z} \qquad\qquad \mathbf{Z} \downarrow \mathbf{Z} \xrightarrow{T_e} \mathbf{Z}$$

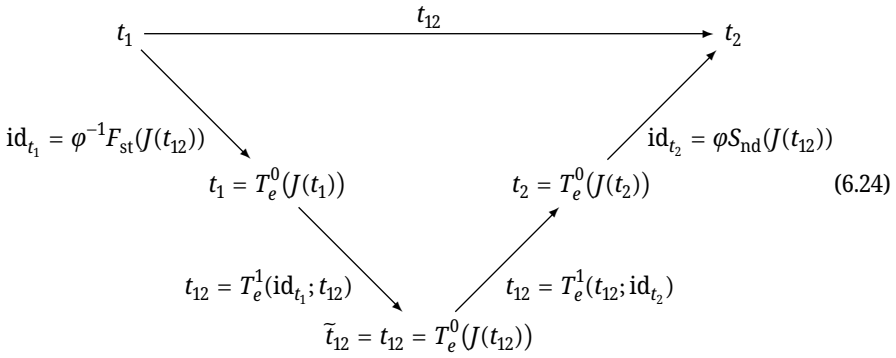
and from the fact that to the identity arrow $\mathrm{id}_{t_2} : t_2 \to t_2$ is associated with the identity path's map $f_{22} = \mathrm{id}_{X_2} : X_2 \to X_2$, and from definition of T_e^1, we obtain

$$\begin{aligned} &T_e^1(t_{12}; \mathrm{id}_{t_2}) = (X_1, \mathrm{ext}_1, \mathrm{agr}_2 \cdot (f_{22} \times f_{22})) = (X_1, \mathrm{ext}_1, \mathrm{agr}_{12}), \quad \text{and from (6.21),} \\ &\tau^{-1}(J(t_{12})) = T_e^1(t_{12}; \mathrm{id}_{t_2}) = t_{12}. \end{aligned} \tag{6.23}$$

Thus,

$$\begin{aligned} (\tau^{-1} \bullet \tau)(J(t_{12})) &= \tau^{-1}(J(t_{12})) \circ \tau(J(t_{12})) \\ &= t_{12} \circ t_{12} = (X_1, \mathrm{ext}_1, \mathrm{agr}_{12} \cdot (\mathrm{id}_{X_1} \times \mathrm{id}_{X_1})) \\ &= (X_1, \mathrm{ext}_1, \mathrm{agr}_{12}) = t_{12} \\ &= \psi(J(t_{12})) \end{aligned}$$

i. e., $\tau^{-1} \bullet \tau = \psi$, and hence, $\mathbf{Z}$ is a symmetry-extended category, with the following detailed diagram for each arrow $t_{12} : t_1 \to t_2$:



(6.24)

which is equal to the commutative diagram in (6.13). □

Consequently, in each symmetry-extended lp 2-category, the 2-cells are the ordinary 1-cells as well, and we can use only the category $\mathbf{Z}$ to express both compositions of 1-cells and 2-cells, without necessity to use the poset categories $\mathbf{Z}(t_1, t_2)$ for each pair of objects t_1 and t_2 of $\mathbf{Z}$. This result can be extended to symmetry extended bicategories as well.

We can also introduce the tensorial product, by the associative functor $\otimes : \mathbf{Z} \times \mathbf{Z} \to \mathbf{Z}$, such that for each two objects (trees) $t_1 = (X_1, \mathrm{ext}_1, \mathrm{agr}_1)$ and $t_2 = (X_2, \mathrm{ext}_2, \mathrm{agr}_2)$ of $\mathbf{Z}$, we define the object $t_1 \otimes t_2 = (X_1 \times X_2, \mathrm{ext}, \mathrm{agr})$, such that $\mathrm{ext}(x_i, y_j) = \mathrm{ext}_1(x_i) \centerdot \mathrm{ext}_2(y_j)$ and $\mathrm{agr}((x_i, y_j), (x_i', y_j')) = \mathrm{ext}_1(x) \centerdot \mathrm{agr}_2(y_j, y_j')$ if $x = x'$; $\mathrm{agr}_1(x_i, x_i')$ otherwise.

Intuitively, $t_1 \otimes t_2$ is the sequential composition, by attaching a copy of t_2 at every leaf of t_1. Thus, a path in $t_1 \otimes t_2$ is a path in t_1 followed by a path in t_2. If we denote an empty tree by $t_\emptyset$ composed by the root node only and extension an agreement empty functions, then it is a unit element for the monoidal composition, i. e., for any tree t, $t \otimes t_\emptyset = t_\emptyset \otimes t = t$. Note that $t_\emptyset$ is initial object in $\mathbf{Z}$.

6.4 Internal symmetry of the category of relations

We introduce the category **Rel** and its higher n-dimensional levels in order to be able to represent the transition systems and strong bisimulation as the greatest fixed points in the n-dimensional level $\mathbf{Rel}_3 = (\mathbf{Rel} \downarrow \mathbf{Rel}) \downarrow (\mathbf{Rel} \downarrow \mathbf{Rel})$. The structure of trees obtained as an unfolding of LTS will be presented in a different way from the finite labeled trees [48] given by Definition 30: Instead of a finite, monoid labeled by transitions in $(A^*, \centerdot, \epsilon, \leq, \wedge)$, where tree $t = (X, \text{ext}, \text{agr})$ is a set X of paths, equipped by two functions extension and agreement, here the structure of a finite tree will be given by monoid-labeled *nodes* in the monoid N^* of natural numbers in N with unit ϵ (empty string), so that the root node is labeled by ϵ and other by the strings of natural numbers in N^*.

In this section, we will generalize the conceptual (internal) categorial symmetry properties that are valid for the **Set** category, shown in the Example 9 in Section 3.2, in Corollary 6 in Section 3.3, and in Example 10 in Section 3.4 where it is shown that **Set** category is a symmetry-extended category: given any function $f : a \longrightarrow b$, the conceptualized object of this function is the graph of this function (which is a set),

$$\widetilde{f} = B_{\top}(f) \triangleq \{(x, f(x)) \mid x \in a\}.$$

The composition of objects $*$ in **Set** is defined as an associative composition of binary relations (the graphs of functions), $B_{\top}(g \circ f) = \{(x, (g \circ f)(x)) \mid x \in a\} = \{(y, g(y)) \mid y \in b\} \circ \{(x, f(x)) \mid x \in a\} = B_{\top}(g) * B_{\top}(f)$.

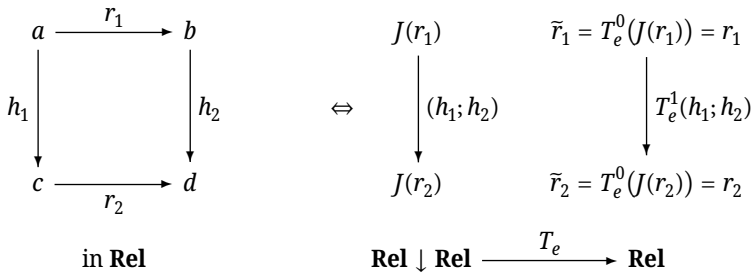
Let us show that this results are valid for a more general category of relations **Rel** (w. r. t. the **Set** category) where the arrows are the binary relations between the sets (i. e., the objects of **Rel**) instead of functions as in the **Set** category.

Proposition 16. *The category* **Rel** *is a symmetry-extended category, conceptually "closed" by the symmetry-closure functor* $T_e = (T_e^0, T_e^1) : \mathbf{Rel} \downarrow \mathbf{Rel} \to \mathbf{Rel}$ *such that:*

1. *The object component of this functor is* $T_e^0 = \psi = J^{-1}$ *where* J^{-1}*, defined by* (1.2)*, is the inverse of the encapsulation operation* J *such that for each object* $J(r_i)$ *of the binary relation* r_i *(an arrow in* **Rel***), we have that* $T_e^0(J(r_i)) = r_i$*, and hence, the conceptualized object obtained from the arrow (relation)* r_i *is the set* $\widetilde{r}_i = r_i$*.*
2. *The arrow component of this functor* T_e^1 *is defined for each arrow* $(h_1; h_2) : J(r_1) \to J(r_2)$ *in* **Rel** $\downarrow$ **Rel** *by*

$$T_e^1(h_1; h_2) \triangleq \{((x, y), (z, w)) \mid (x, y) \in r_1, (x, z) \in h_1,\ (z, w) \in r_2, (y, w) \in h_2\}. \quad (6.25)$$

Proof. It is easy to show that **Rel** is a Conceptually Closed Category (CoCC) by the covariant descending functor $T_e = (T_e^0, T_e^1) : \mathbf{Rel} \downarrow \mathbf{Rel} \to \mathbf{Rel}$, such that $T_e \circ \blacktriangle \simeq I_{\mathbf{Rel}} : \mathbf{Rel} \to \mathbf{Rel}$. In fact, let us consider the following commutative diagram in **Rel**:



(6.26)

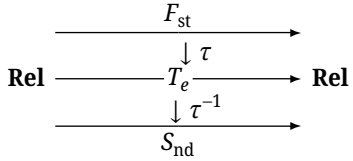
The natural isomorphism (3.13), $\varphi : T_e \circ \blacktriangle \simeq I_{\mathbf{Rel}}$, serves to guarantee that each object (set) a of **Rel** is isomorphic to the conceptual object $\widetilde{\mathrm{id}}_a = T_e(\mathrm{id}_a) = \mathrm{id}_a$ obtained from the identity arrow (a binary relation), i. e.,

$$\widetilde{\mathrm{id}}_a = \mathrm{id}_a = \{(x,x) \mid x \in a\}, \tag{6.27}$$

and hence, we have the bijection $\mathrm{is}_a : \widetilde{\mathrm{id}}_a \to a$ such that for each $(x,x) \in \widetilde{\mathrm{id}}_a$, $\mathrm{is}_a(x,x) = x$ and $\mathrm{is}_a^{-1}(x) = (x,x)$. Thus, the conceptualized object (a set) $\widetilde{\mathrm{id}}_a$ (a binary relation) obtained from the identity arrow id_a (a binary relation which, in this case, is just the graph of the identity function of the set a) is isomorphic to the object a in **Rel**. Thus, the isomorphism $\varphi(a) : \widetilde{\mathrm{id}}_a \to a$ is the following arrow (a binary relation) in **Rel**:

$$\varphi(a) = \{((x,x),x) \mid x \in a\} \quad \text{and} \quad \varphi^{-1}(a) = \{(z,(z,z)) \mid z \in a\}. \tag{6.28}$$

So, from the fact that **Rel** is a CoCC, we have the natural transformations (3.15), $\tau = (T_e\sigma) \bullet (\varphi^{-1}F_{\mathrm{st}})$ and $\tau^{-1} = (\varphi S_{\mathrm{nd}}) \bullet (T_e\sigma^{-1})$, shown in the following diagram:



with their vertical composition (3.16), $\tau^{-1} \bullet \tau = (\varphi S_{\mathrm{nd}}) \bullet (T_e(\psi;\psi)) \bullet (\varphi^{-1}F_{\mathrm{st}})$. Let us show that $\tau^{-1} \bullet \tau = \psi$.

For any arrow (a binary relation) $r : a \to b$ of **Rel**, and the arrow $(r,r) : J(\mathrm{id}_a) \to J(\mathrm{id}_b)$ of **Rel** $\downarrow$ **Rel**, from (6.25) we have that

$$T_e^1(r;r) = \{((x,y),(z,w)) \mid (x,y) \in \mathrm{id}_a, (x,z) \in r,\ (z,w) \in \mathrm{id}_b, (y,w) \in r\}$$

so that it must hold, from (6.27), that $y = x$ and $w = z$, and hence,

$$T_e^1(r;r) = \{((x,x),(z,z)) \mid (x,z) \in r\}. \tag{6.29}$$

Thus, for the object $J(r)$ of **Rel** $\downarrow$ **Rel**, we obtain

$$\begin{aligned}
\tau^{-1} \bullet \tau(J(r)) &= ((\varphi S_{\mathrm{nd}}) \bullet (T_e(\psi;\psi)) \bullet (\varphi^{-1} F_{\mathrm{st}}))(J(r)) \\
&= (\varphi S_{\mathrm{nd}})(J(r)) \circ (T_e(\psi;\psi))(J(r)) \circ (\varphi^{-1} F_{\mathrm{st}})(J(r)) \\
&= \varphi(b) \circ T_e(r;r) \circ \varphi^{-1}(a) \\
&= \varphi(b) \circ \{((x,x),(z,z)) \mid (x,z) \in r\} \circ \varphi^{-1}(a) \quad \text{from (6.29)} \\
&= \{(x,z) \mid (x,z) \in r\} \quad \text{from (6.28)} \\
&= r \\
&= \psi(J(r))
\end{aligned}$$

i. e., $\tau^{-1} \bullet \tau = \psi$, and hence, from Definition 14 in Section 3.4, **Rel** is a symmetry-extended category. □

In Section 2.4, we considered the comma-propagation of the limits. Let us now consider the comma-propagation of the particular case of the limits: the *pullbacks*.

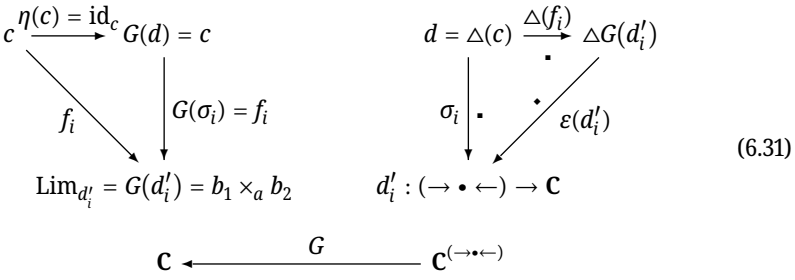
In this case for the limits, we use the adjunction $(\triangle, G, \varepsilon, \eta)$ presented in the proof of Theorem 4 in Section 2.3, where functor $G = (G^0, G^1) : \mathbf{C}^{\mathbf{J}} \to \mathbf{C}$ is the right adjoint to the general diagonal functor $\triangle : \mathbf{C} \to \mathbf{C}^{\mathbf{J}}$ and the small index category $\mathbf{J} = (\to \bullet \leftarrow)$ is the category composed by three indexed-objects a_1, a_2 and a_3 with their identity arrows, and with two nonidentity arrows having common codomain object a_3.

The functor G is defined by the following: For each object (a functor) $d = (d^0, d^1) : (\to \bullet \leftarrow) \to \mathbf{C}$ and arrow (a natural transformation) $\sigma_i : d \xrightarrow{\cdot} d_i'$ in $\mathbf{C}^{(\to\bullet\leftarrow)}$, the "fibered product" (or product over the object $d^0(a_3)$ in $\mathbf{C}$),

$$G^0(d) \triangleq d^0(a_1) \times_{d^0(a_3)} d^0(a_2) \tag{6.30}$$

so that for any object $c \in \mathrm{Ob}_{\mathbf{C}}$ and the object (functor) d_i' in $\mathbf{C}^{(\to\bullet\leftarrow)}$ representing a diagram with $b_1 = {d_i'}^0(a_1)$, $b_2 = {d_i'}^0(a_2)$ and $a = {d_i'}^0(a_3)$ for which we define the limit cone with its vertex equal to the created limit (fibered product) $G^0(d_i') = b_1 \times_a b_2$ in $\mathbf{C}$.

So, such diagrams are represented by the category of functors $\mathbf{C}^{(\to\bullet\leftarrow)}$, and let $G : \mathbf{C}^{(\to\bullet\leftarrow)} \to \mathbf{C}$ be the functor, which creates the limits for such diagrams, and hence, right adjoint to the general diagonal functor $\triangle : \mathbf{C} \to \mathbf{C}^{(\to\bullet\leftarrow)}$, so that this adjunction is represented by the following commutative diagrams:



(6.31)

where the functor $d = \triangle(c)$ represents the diagram $c \xrightarrow{\mathrm{id}_c} c \xleftarrow{\mathrm{id}_c} c$, from the fact that $d^0(a_i) = \triangle(c)(a_i) = c$ and $d^1(l_{32}) = d^1(l_{12}) = \triangle(c)(l_{12}) = \mathrm{id}_c$, and hence, can be reduced into the object c. That is, the limit of this diagram is

$$G(d) = c \tag{6.32}$$

so that $\eta(c) = \mathrm{id}_c$, and hence, $G(\sigma_i) = f_i$ in $\mathbf{C}$, and hence, the commutative diagram in the left-hand side above reduces to the simple arrow $f_i : c \to \mathrm{Lim}_{d_i'}$. For the commutative diagram in the right-hand side in $\mathbf{C}^{(\rightarrow\bullet\leftarrow)}$ from the vertex c into diagram $b_1 \xrightarrow{r_1} a \xleftarrow{r_2} b_2$ (for $b_1 = {d_i'}^0(a_1)$, $b_2 = {d_i'}^0(a_2)$, $a = {d_i'}^0(a_3)$, $r_1 = {d_i'}^1(l_{13})$ and $r_2 = {d_i'}^1(l_{23}')$), we have that for a cone σ_i defined by $\sigma_i(a_1)$ into the limit cone $(\triangle G(d_i'))(a_1) = G(d_i') = \mathrm{Lim}_{d_i'}$ defined by the arrows $p_j = (\varepsilon(d_i'))(a_j)$. For $j = 1, 2$, there is the arrow $f_i = (\triangle(f_i))(a_1)$, which makes this diagram commutative, and hence, from it we obtain the pullback commutative diagram in $\mathbf{C}$,

$$\begin{array}{ccccc} c = (\triangle(c))(a_1) & & \xrightarrow{f_i = (\triangle(f_i))(a_1)} & & (\triangle G(d_i'))(a_1) = \mathrm{Lim}_{d_i'} \\ & \searrow^{\sigma_i(a_1)} & & \swarrow_{p_1} \quad \searrow^{p_2} & \\ & & b_1 \xrightarrow{r_1} a \xleftarrow{r_2} & & b_2 \end{array} \tag{6.33}$$

where the commutative triangle $r_1 \circ p_1 = r_2 \circ p_1$ is the well-known pullback square diagram. Let us show that we have such pullback square diagrams in the case when $\mathbf{C}$ is the **Rel** category: in order to use the pullbacks for transition systems, we will use the category with sets as objects and binary relations as arrows.

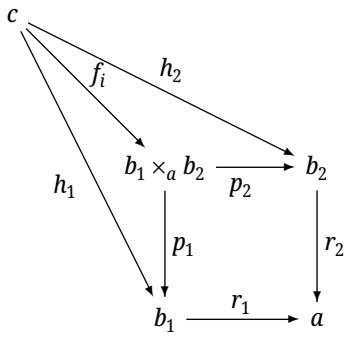
Proposition 17. *There exists the following pullback square diagram in the* **Rel** *category with sets as objects and binary relations as arrows, for each two arrows (binary relations) $r_1 : b_1 \to a$ and $r_2 : b_2 \to a$ of the given diagram $d' : (\rightarrow \bullet \leftarrow) \to$* **Rel***:*

$$\begin{array}{ccc} b_1 \times_a b_2 & \xrightarrow{p_2} & b_2 \\ \downarrow{\scriptstyle p_1} & & \downarrow{\scriptstyle r_2} \\ b_1 & \xrightarrow[r_1]{} & a \end{array} \tag{6.34}$$

where p_1 and p_2 are the first and second projection (cone arrows of the pullback limit object) and the fibered product (pullback) is defined by

$$b_1 \times_a b_2 \triangleq \{(x, y) \mid \exists z \in a, (x, z) \in r_1, (y, z) \in r_2\}. \tag{6.35}$$

Proof. Let us consider, for a given object c, the commutative diagram (6.33), by the arrows $h_j = \sigma_i(a_j) : c \to b_j$, for $j = 1, 2$, such that $r_1 \circ h_1 = r_1 \circ h_1$, so that the commutative diagram (6.33) can be presented by the following usual pullback square diagram:



(6.36)

and let us show that f_i is the unique arrow from the cone with the vertex c into the limit cone with the pullback vertex $b_1 \times_a b_2$. The projection relations p_1 and p_2 are defined by

$$p_1 = \{((y_i, y_j), y_i) \mid (y_i, y_j) \in b_1 \times_a b_2\} \quad \text{and} \quad p_2 = \{((y_i, y_j), y_j) \mid (y_i, y_j) \in b_1 \times_a b_2\}. \tag{6.37}$$

Suppose that there exists another arrow $k : c \to b_1 \times_a b_2$, different from f_i, such that the diagram above commutes, i. e., with

$$p_1 \circ k = h_1 \quad \text{and} \quad p_2 \circ k = h_2. \tag{6.38}$$

In that case must exist some $w \in c$, such that

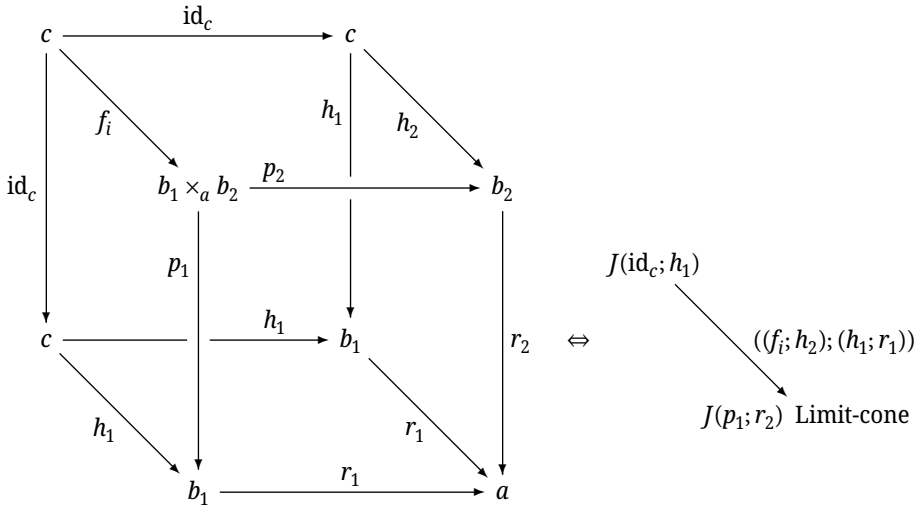
$$\{x \mid (w, x) \in k\} \neq \{x \mid (w, x) \in f_i\}. \tag{6.39}$$

Let $x' = (y_1, y_2)$ be an element of fibered product $b_1 \times_a b_2$, such that $(w, (y_1, y_2)) \in k$ but $(w, (y_1, y_2)) \notin f_i$. Consequently, from (6.38) it must hold that $(w, y_1) \in h_1$ and $(w, y_2) \in h_2$, while from (6.39) it must be that $(w, y_1) \notin h_1$ or $(w, y_2) \notin h_2$, which is a contradiction.

Thus, f_i is unique arrow (a binary relation) in **Rel** for which the diagram above commutes, and hence, the fiber product is the limit object (pullback). □

We have shown the comma-propagation of all limits in all n-dimensional levels in Section 2.3.

The pullback commutative diagrams in (6.36) can be represented geometrically by the following *3-dimensional* cube in **Rel**:



(6.40)

represented equivalently in the n-dimensional level $\mathbf{Rel}_3$ by the single arrow $((f_i;h_2);(h_1;r_1)) : J(\mathrm{id}_c;h_1) \to J(p_1;r_2)$ where the codomain object $J(p_1;r_2)$ represents the limit cone of the pullback front square commutative diagram of this cube, while the domain object $J(\mathrm{id}_c;h_1)$ (the back square diagram of this cube) represents a cone with vertex in object c of **Rel**.

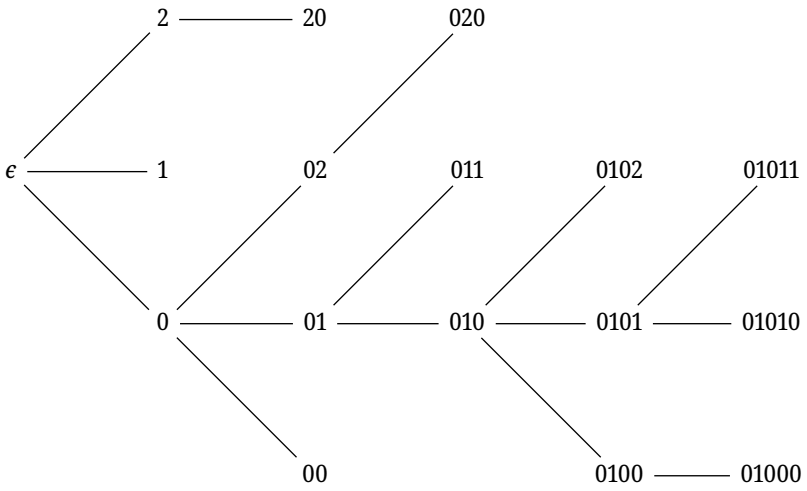
6.5 Transition systems as fixed points in the *n*-dimensional level $\mathbf{Rel}_3$

In what follows, we will use a more general definition of the finite trees, based on the labeling of its nodes only by the strings of a monoid $(N^*, \centerdot, \epsilon)$ of natural numbers in N with unity ϵ equal to empty string. So, a finite tree based on this monoid, denoted by the finite prefix-closed subset $M \subseteq N^*$, such that $\epsilon \in M$ and if the finite string $n_1 n_2 \dots n_k \dots \in M$ then also $n_1 n_2 \dots n_k \in M$, for $n_i \in N$, is represented here in an analog way of definition of synchronization trees [43, 47].

We will show also how such an abstract definition of the finite trees can be enriched, in the way to obtain from them the finite labeled trees introduced by Definition 30 and used in the previous sections.

A concrete method of definition of such abstract finite trees with nodes labeled by monoid N^*, both with the introduction of the fundamental function $f_N : N^* \to N^*$, can be shown by the following example.

Example 18. Let us consider the following tree defined by its nodes labeled by the strings in N^* so that the label of each node, with its father labeled by the string v, is equal to $v \cdot n$, where $n \in N$ begins from 0 and increments from down to up. In this way, the length of the label of each node corresponds to the level of this node in this tree, i. e., to the length of the path from this node to the root of the tree:



Thus, this tree is defined by the set of its labeled nodes,[3]

$$M = \{\epsilon, 0, 1, 2, 00, 01, 02, 20, 010, 011, 020, 0100, 0101, 0102, 01000, 01010, 01011\} \subset N^*,$$

with the names $x_i \in X$ of the paths (equal to the labels of leaf nodes of the trees), so that we have nine paths $X = \{1, 2, 00, 011, 020, 0102, 01000, 01010, 01011\} \subset N^*$.

The movement from a given node to its father node is defined by the function

$$f_N : N^* \to N^* \tag{6.41}$$

such that for any $n_1 \dots n_{k-1} n_k \in N^*$ with $n_i \in N$ for all $1 \leq i \leq k$ with $k \geq 2$,

$$f_N(n_1 \dots n_{k-1} n_k) \triangleq n_1 \dots n_{k-1} \quad \text{and} \quad f_N(n_i) = f_N(\epsilon) = \epsilon \tag{6.42}$$

is the node label of the father in this tree. By repeating this function a finite time, we reach the root ϵ of the tree (the fixed point). For example, in the tree above we obtain

3 Note that here we denoted the strings in compact way, but the correct formal denotation has to divide the components of the string by the symbol of composition $\centerdot$ of the monoid $(N^*, \centerdot, \epsilon)$. That is, e. g., the string 01011 composed by 5 numbers has to be denoted correctly by $0 \centerdot 1 \centerdot 0 \centerdot 1 \centerdot 1$, to distinguish it from the string $0 \centerdot 1 \centerdot 0 \centerdot 11$ composed by 4 numbers, or from the string $0 \centerdot 1011$ composed by only 2 numbers in N.

$f_N^5(01010) = f_N^3(011) = f_N^2(20) = f_N(1) = \epsilon$. If, for a given label v of a node, we denote by $|v|$ the length of this string, then we obtain that $f_N^{|v|} = \epsilon$.

Based on this method of labeling the nodes of an abstract finite tree M, provided in the example above, such that $M \subseteq N^*$ is a finite prefix-closed subset, we can show how we can associate it to a transition labeled system $\mathrm{TS} = (S, i_0, A, \mathrm{Tran})$ by an enrichment of this abstract tree by two new functions: a function $f_A : M' \to A$, where M' is the set M without ϵ, for labeling its arcs (transitions in Tran of this LTS), and a function $f_S : M \to S$ for labeling the states of this LTS.

Corollary 14. *Let M be a finite subset of N^* with two functions:*
- *$f_S : M \to S$, which associate the states in S to the strings in M.*
- *Let $M' \subseteq M$ be the subset without ϵ, and $f_A : M' \to A$ be a function, which associate actions in A and the elements in M'.*

If we define, by using the function $f_N : N^* \to N^*$ in (6.42)*, two relations*

$$r_1 = \{((f_S f_N(x), f_S(x)), x) \mid x \in M'\} \quad \textit{and} \quad r_2 = \{(f_A(x), x) \mid x \in M'\} \tag{6.43}$$

then the following diagram is a pullback diagram in **Rel** *with $S^2 = S \times S$,*

$$\begin{array}{ccc} S^2 \times_{N^*} A & \xrightarrow{p_2} & A \\ \downarrow{\scriptstyle p_1} & & \downarrow{\scriptstyle r_2} \\ S^2 & \xrightarrow[r_1]{} & N^* \end{array} \tag{6.44}$$

where the pullback object (a fibered product) is defined by

$$S^2 \times_{N^*} A \triangleq \{((s_i, s_j), a) \mid ((s_i, s_j), x) \in r_1 \quad \textit{and} \quad (a, x) \in r_2\} \tag{6.45}$$

with projection arrows

$$\begin{aligned} p_1 &\triangleq \{(((s_i, s_j), a), (s_i, s_j)) \mid ((s_i, s_j), a) \in S^2 \times_{N^*} A\} \\ p_2 &\triangleq \{(((s_i, s_j), a), a) \mid ((s_i, s_j), a) \in S^2 \times_{N^*} A\}. \end{aligned} \tag{6.46}$$

Proof. It is easy to verify the following the proof in Proposition 17. □

Notice that in the definition above of a finite subset M (nonnecessarily prefix-closed), M represents more than one different tree. Thus, in order to represent a single tree (obtained as unfolding of a given LTS) the subset M used in the pullback diagram in (6.44) must be prefix-closed: for example, by using the method in Example 18.

Proposition 18. *Let $M \subseteq N^*$ be a finite prefix-closed subset with the pullback diagram* (6.44) *of Corollary* 14. *Let us define the function $f : N^* \to N^*$ defined for each $x \in N^*$ by*

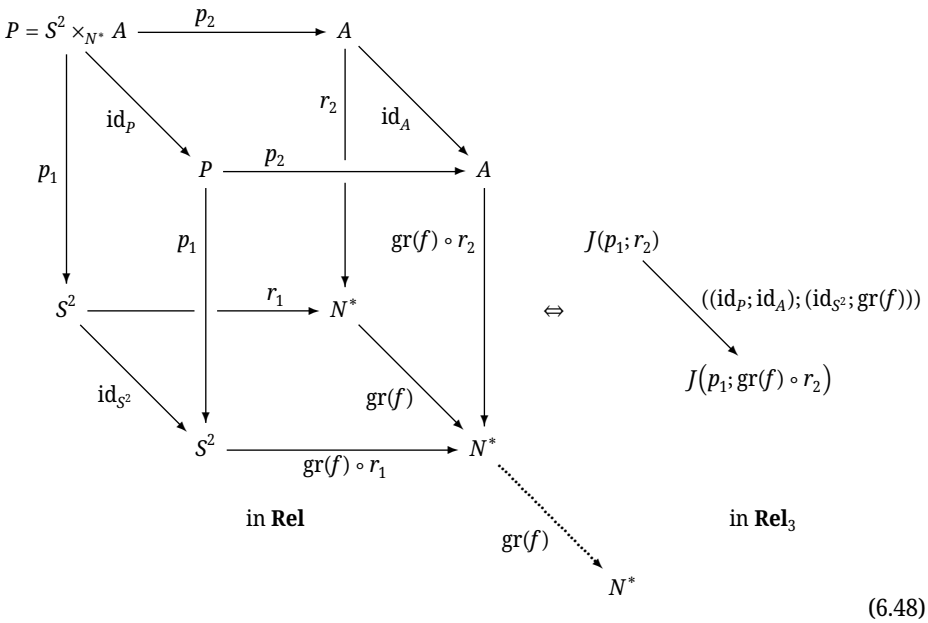
$$f(x) = f_N(x) \quad \text{if } x \neq \epsilon \text{ and } f_N(x) \in M;\ x \text{ otherwise} \tag{6.47}$$

with its graph (a binary relation) $\mathrm{gr}(f)$. Then, for the pullback $P = S^2 \times_{N^*} A$, the quadruple of binary relations $((\mathrm{id}_P; \mathrm{id}_A); (\mathrm{id}_{S^2}; \mathrm{gr}(f)))$, which is an arrow in $\mathbf{Rel}_3$, has the fixed-point object $n_f = J(p_1; \mathrm{gr}(f^n) \circ r_2)$ in $\mathbf{Rel}_3$, for $n = \max\{|x| \mid x \in M\}$, and $\mathrm{gr}(f^n)$ is the graph (a binary relation) of the function $f^n = \overbrace{f \cdot \cdots \cdot f}^{n} : N^* \to N^*$.

This pullback P defines the rooted labeled transition system $\mathrm{TS} = (S, i_0, A, \mathrm{Tran})$, *such that from Definition* 26,

- *Transition relation,* $\mathrm{Tran} = \{(s_i, a, s_j) \mid ((s_i, s_j), a) \in S^2 \times_{N^*} A\}$
- *Initial state, $i_0 = f_S \pi_2 S^1_{\mathrm{nd}} \psi(n_f)$*

Proof. Let us consider the following commutative diagram in **Rel**:



(6.48)

where by iteration in $\mathbf{Rel}_3$ n-times of the arrow $((\mathrm{id}_P; \mathrm{id}_A); (\mathrm{id}_{S^2}; \mathrm{gr}(f)))$ we obtain the arrow $((\mathrm{id}_P; \mathrm{id}_A); (\mathrm{id}_{S^2}; \mathrm{gr}(f^n))) : J(p_1, r_2) \to n_f$, with $n_f = J(p_1; \mathrm{gr}(f^n) \circ r_2)$. Note that $\mathrm{gr}(f^n) = \{(x, f^n(x)) \mid x \in M\} = \{(x, \epsilon) \mid x \in M\}$ from the fact that for each string $x \in M$, $f^n(x) = \epsilon$ for $n = \max\{|x| \mid x \in M\}$ (the fixed point for a tree is its root).

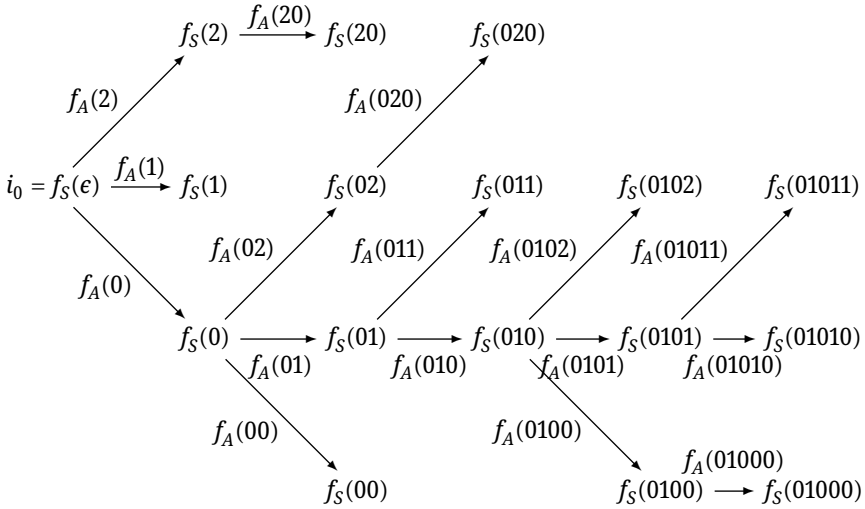
Thus, we obtain the binary relation $\mathrm{grf}(f^n) \circ r_2 = \{(f_A(x), \epsilon) \mid x \in M \text{ and } x \neq \epsilon\}$, and hence,

$$f_S \pi_2 S^1_{\mathrm{nd}} \psi(n_f) = f_S \pi_2 S^1_{\mathrm{nd}} \psi(J(p_1; \mathrm{grf}(f^n) \circ r_2))$$

$$= f_S\pi_2(\mathrm{grf}(f^n) \circ r_2) = f_S\pi_2(\{(f_A(x), \epsilon) \mid x \in M \text{ and } x \neq \epsilon\})$$
$$= f_S(\epsilon) = i_0,$$

is the initial state (of the root ϵ of the tree M). □

It is easy to verify that the transition relation Tran of the LTS corresponding to the abstract tree M is well-defined, such that for each node $x \in M$, the function $f_S : M \to S$ defines the state while f_A defines the label (action) of the ingoing transition for this state. So, the unfolding of the LTS, obtained by the abstract tree M in Example 18, is the tree:



Such trees, obtained from the unfolding of LTS derived from the abstract tree structure M by using Proposition 18, can be used to define the finite labeled trees $(X, \mathrm{ext}, \mathrm{agr})$ given in Section 6.2 by Definition 30, as follows:

– The set of paths X is defined by the set of leafs of the abstract tree M. For the example above, it is the subset

$$\{1, 00, 20, 011, 020, 0100, 0102, 01000, 01010, 01011\} \subset M.$$

– The function $\mathrm{ext} : X \to A^*$ is defined for each $x = n_1 \ldots n_k$ in X by $\mathrm{ext}(x) = f_A(n_1) \ldots f_A(n_k)$.
– The agreement $\mathrm{agr} : X \times X \to A^*$ is defined for any two paths $x, y \in X$ by $\mathrm{agr}(x, y) = \mathrm{ext}(x) \wedge \mathrm{ext}(y)$, as usual.

Consequently, the results used in this section can be also used for the symmetry-extended 2-category **Z** of Definition 31 in Section 6.3.

6.6 Strong bisimulations as fixed points in the n-dimensional level $\mathbf{Rel}_3$

Many of the proposed equivalences are based on the notion of bisimulation [41] that gives rise to the so-called *strong* observational equivalence over LTS. Two systems are considered as equivalent if they can perform the same labeled transitions to reach equivalent states. It leads to basic behavioral abstraction and, in the CCS languages, to elegant and simple laws [38]. While there is a general agreement about strong observational equivalence as providing the minimum of abstraction required for modeling nondeterministic systems, there is yet a discussion about its generalization. From this fact and our main consideration of a conceptual categorial symmetry and its applications, in this book we will reduce our considerations only to this basic strong observational equivalence.

Bisimulation is a rich concept, which appears in various areas of theoretical computer science. Besides its origin for understanding concurrency, it was independently developed in the context of modal logic. Many logics relevant to computer science are fragments of first-order logic, e. g., k-variable logics and various modal logics. The fundamental semantic notion in basic modal logic is truth of a formula at a state in a Kripke structure; this notion is local and of a first-order nature. Since Kripke models are a special case of (labeled) state transition systems, bisimulation is also a topic in modal logic. In regard to the coherence of the family of modal logics, it is important to understand in model theoretic terms what it is that makes a logic "modal." The mathematics of modality has been applied in theoretical computer science to the category-theoretic notion of a coalgebra, defined in the commutative diagram (A.4) in the Appendix, Section A.2. Any modal frame for a modal logic with a unique existential modal operator $\Diamond$ can be viewed as a coalgebra for the powerset functor $F = \mathcal{P} : \mathbf{Set} \to \mathbf{Set}$. A $\mathcal{P}$-coalgebra $h : S \to \mathcal{P}(S)$ for a given set of states ("worlds") defines a binary accessibility relation R on the set S by

$$(s_i, s_j) \in R \text{ iff } s_j \in h(s_i),$$

giving the frame (S, R), with $h(s_i) = \{s_j \mid (s_i, s_j) \in R\} \in \mathcal{P}(S)$. But this last equation can also be read as a definition of h given R, so there is an exact correspondence between frames and $\mathcal{P}$-coalgebras. Moreover, a function $f_1 : A \to B$ is a coalgebraic morphism (see diagram (A.4) in the Appendix, Section A.2) from the coalgebra $h_1 : A \to F(A)$ to coalgebra $k_1 : B \to F(B)$ precisely when it is a *p-morphism* between the corresponding frames.

Let $\mathfrak{F} = (S, \{R_\alpha\}_{\alpha \in A})$ be the Kripke frame of a multimodal propositional logic with the set Φ of atomic propositions (propositional symbols) $p_i \in \Phi$, and with the set of existential modal operators $\Diamond_\alpha$ for $\alpha \in A$ with corresponding accessibility relations $R_\alpha \subseteq S \times S$. We denote by $\mathcal{M} = (\mathfrak{F}, v)$ a Kripke model for a valuation $v : \Phi \to \mathcal{P}(S)$ such that

for each atomic proposition p_i, $v(p_i) \subseteq S$ is the subset of states ("worlds") where p_i is satisfied, i. e., $v(p_i) = \{s \in S \mid \mathcal{M} \models_s p_i\}$. So, let us consider an example.

Example 19 (Hennessy–Milner logic). Matthew Hennessy and Robin Milner [39] applied modal logic to process algebra in a manner that is reminiscent of the Kripke modeling of PDL. Here, we illustrate with the case of a collection $\{R_\alpha \mid \alpha \in A\}$ of accessibility observation relations associated with the Hennessy–Milner logic. This can be viewed as a coalgebra for a functor $(\mathcal{P}_)^A$ that takes each set S to the set $(\mathcal{P}(S))^A$ of all functions from the set of labels $\alpha \in A$ (each one for some existential modal operator $\Diamond_\alpha$) to $\mathcal{P}(S)$. Using the notation $s_i \xrightarrow{\alpha} s_j$ in place of $(s_i, s_j) \in R_\alpha$, we find that a system $\{R_\alpha \mid \alpha \in A\}$ of relations on a set S corresponds to the coalgebra $h : S \to (\mathcal{P}(S))^A$ for which $h(s_i)$ is the function $\alpha \mapsto \{s_j \mid s_i \xrightarrow{\alpha} s_j\}$.

Definition 32. A coalgebra $h : S \to (\mathcal{P}(S))^A$ for the functor $(\mathcal{P}_)^A : \mathbf{Set} \to \mathbf{Set}$ can be regarded as providing the state-transition relation for a nondeterministic automaton with input set of labels (logical "actions") A and state set S. For each state s_i in S, $h(s_i)(\alpha)$ is the set of possible next states that can be reached by making a transition from s_i on label α.

For this reason, the h-arrow of any kind of coalgebra is often called a (labeled) *transition system* TS, and its domain is thought of as a *state* set. (We can identify (S, h) with its transition structure, since S is determined as the domain of h.)

The theory makes significant use of a definition of bisimulation for coalgebras that was introduced in [49]. A relation $R \subseteq S \times S'$ is a bisimulation from a coalgebra $h : S \to F(S)$ to $k : S' \to F(S')$ when there exists a transition structure $t : R \to F(R)$ on R such that the projection functions from (R, t) to (S, h) and (S', k) are coalgebraic morphisms. There is always a largest such bisimulation, known as the bisimilarity relation from (S, h) to (S', k). We can model finitely branching nondeterminism by using the finitary powerset functor $\mathcal{P}_f$, so that we have also final (or terminal) $(\mathcal{P}_f_)^A$ colagebra as well.

Devising a suitable syntax and semantics for F-coalgebras is a matter that depends on the nature of the functor F involved. A natural desideratum is a satisfaction relation $h \models_x \phi$, expressing 'formula ϕ is true/satisfied at state x in coalgebra h," that provides a logical characterization of bisimilarity in the following form:

$$x \approx_s y \quad \text{iff for all formulae} \quad \phi,\ h \models_x \phi \text{ iff } k \models_y \phi$$

If this holds, we will say that the logic, or the functor F, has the *Hennessy–Milner* (HM) property.

The coalgebraic framework for a definition of the bisimulation for the Hennessy–Milner logic of the example above can be defined also in a more specific modal logic framework, by using Kripke models of a multimodal propositional logic $\mathcal{L}$ with a set of existential modal operators $\Diamond_\alpha$ for $\alpha \in A$.

Definition 33. Given two Kripke models $\mathcal{M}_1 = (S_1, \{R_\alpha\}_{\alpha\in A}, v_1)$ and $\mathcal{M}_2 = (S_2, \{R'_\alpha\}_{\alpha\in A}, v_2)$, a bisimulation between $\mathcal{M}_1$ and $\mathcal{M}_2$ is a nonempty relation $R \subseteq S_1 \times S_2$ satisfying the following condition for any $(s_j, s) \in R$:

1. *Atom equivalence*: s_j and s satisfy the same atomic proposition, i. e., $v_1(s_j) = v_2(s)$, hereafter denoted by $s_j \approx_s s$.
2. *Forth*: for any label $\alpha \in A$, if $(s_j, s_k) \in R_\alpha$ for some $s_k \in S_1$, then there is some $s' \in S_2$ such that $(s, s') \in R'_\alpha$ and $s_k \approx_s s'$ (any transition $s_j \xrightarrow{\alpha} s_k$ in $\mathcal{M}_1$ can be matched by transition $s \xrightarrow{\alpha} s'$ in $\mathcal{M}_2$).
3. *Back*: Similarly, in the opposite direction: for any $\alpha \in A$, if $(s, s') \in R'_\alpha$ for some $s' \in S_2$, then there is some $s_k \in S_1$ such that $(s_j, s_k) \in R_\alpha$ and $s_k \approx_s s'$ (any transition $s \xrightarrow{\alpha} s'$ in $\mathcal{M}_2$ can be matched by transition $s_j \xrightarrow{\alpha} s_k$ in $\mathcal{M}_1$).

That R is a bisimulation between $\mathcal{M}_1$ and $\mathcal{M}_2$ is denoted by $R : \mathcal{M}_1 \rightleftarrows \mathcal{M}_2$.

In Definition 27, Section 6.1, it was specified that given a LTS, $\mathrm{TS}_k = (S_k, i_{0k}, A, \mathrm{Tran}_k)$ for $k = 1, 2$, a symmetric relation $R \subseteq S_1 \times S_2$ is a strong bisimulation if $(s_j, s) \in R$ and $s_j \xrightarrow{\alpha} s_k$, i. e., $(s_j, \alpha, s_k) \in \mathrm{Tran}_1$ implies that there exists transition $s \xrightarrow{\pi} s'$, i. e., $(s, \alpha, s') \in \mathrm{Tran}_2$, such that $(s_k, s') \in R$.

For example, such finite-branching LTS can be derived by Definition 32, from a coalgebra $h : S \to (\mathcal{P}_f(S))^A$. Thus, based on the Corollary 14 and Proposition 18, for definition of LTS from finite abstracted trees M_k (prefix-closed subsets $M_k \subseteq N^*$) for $k = 1, 2$, we obtain that their transition relations are defined from the pullback objects of the pullback commutative diagrams (6.48) in the category **Rel**. Thus, in what follows we will define the strong bisimulation relation R as a fixed point in the n-dimensional level $\mathbf{Rel}_3$, in a coinductive way from the initial bisimulation subset $R_0 \subset R$ of the bisimilar states that are the leafs of the abstract trees M into the next steps $R_n \subseteq R_{n+1}$ for $n = 0, 1, 2, \ldots$

Proposition 19. *Let, for $k = 1, 2$, $\mathrm{TS}_k = (S_k, i_{0k}, A, \mathrm{Tran}_k)$ be two LTS with their unfolding equal to two finite abstract trees $M_k \subseteq N^*$ with the leafs in $X_k \subset M_k$, with the functions f_{S_k} that define the states of the nodes of M_k and f_{A_k} that define the ingoing transition labels (actions) into these states. So, we define the object R_0 for the strongly bisimilar leafs of these two trees,*

$$R_0 \triangleq \{(x_i, x_j) \mid x_i \in X_1, x_j \in X_2 \quad \textit{and} \quad f_{S_1}(x_i) \approx_s f_{S_2}(x_j)\} \tag{6.49}$$

and two arrows $r_1 : S \to R_0$ and $r_2 : S \to R_0$ in **Rel**,

$$r_1 = \{(f_{S_1}(x_i), (x_i, x_j)) \mid (x_i, x_j) \in R_0\}, \quad r_2 = \{(f_{S_2}(x_j), (x_i, x_j)) \mid (x_i, x_j) \in R_0\} \tag{6.50}$$

so that we obtain the limit cone (pullback square commutative diagram),

$$\begin{array}{ccc} S_1 \times_{R_0} S_2 & \xrightarrow{p_2} & S_2 \\ \downarrow{\scriptstyle p_1} & & \downarrow{\scriptstyle r_2} \\ S_1 & \xrightarrow[r_1]{} & R_0 \end{array} \tag{6.51}$$

with the projections p_1 and p_2 and the pullback object (fibered product),

$$S_1 \times_{R_0} S_2 \triangleq \{(s,s') \mid (x_i,x_j) \in R_0, \quad \textit{such that } (s,(x_i,x_j)) \in r_1 \textit{ and } (s',(x_i,x_j)) \in r_2\}.$$

Proof. It is easy to verify following the proof in Proposition 17. □

Note that the bisimilar leafs of the set R_0 in (6.49) are determined depending on the semantics of the LTS: for example, in the case of LTS derived from the multimodal propositional logics, $R_0 = \{(x_i,x_j) \mid x_i \in X_1, x_j \in X_2 \text{ and } \nu_1(f_{S_1}(x_i)) = \nu_2(f_{S_1}(x_i))\}$ for two Kripke models in Definition 33.

The fixed point is obtained by coinductive iteration of some function f (in diagram (6.48)), with the first step domain equal to R_0 relation of the strongly bisimilar leafs of the trees M_1 and M_2, i. e., starting from the leafs of the trees toward their roots.

Definition 34. Given a subset $R_n \subseteq M_1 \times M_2$, we define the function $f : R_n \to M_1 \times M_2$, such that for any $(x_i,y_j) \in R_n$,

$$f(x_i,x_j) = \big(f_N(x_i), f_N(x_j)\big) \quad \text{if } (\mathbf{a});\ (x_i,x_j) \text{ otherwise} \tag{6.52}$$

where (**a**) is the following condition:

for every $x \in M_1$ such that $f_N(x) = f_N(x_i)$, there exists $x' \in M_2$ such that $f_N(x') = f_N(x_j)$ and $f_{A_1}(x) = f_{A_2}(x')$ and $(x,x') \in R_n$.

If we denote by $f(D_n)$ the image of this function, then we can replace this function by the equivalent to it, the function

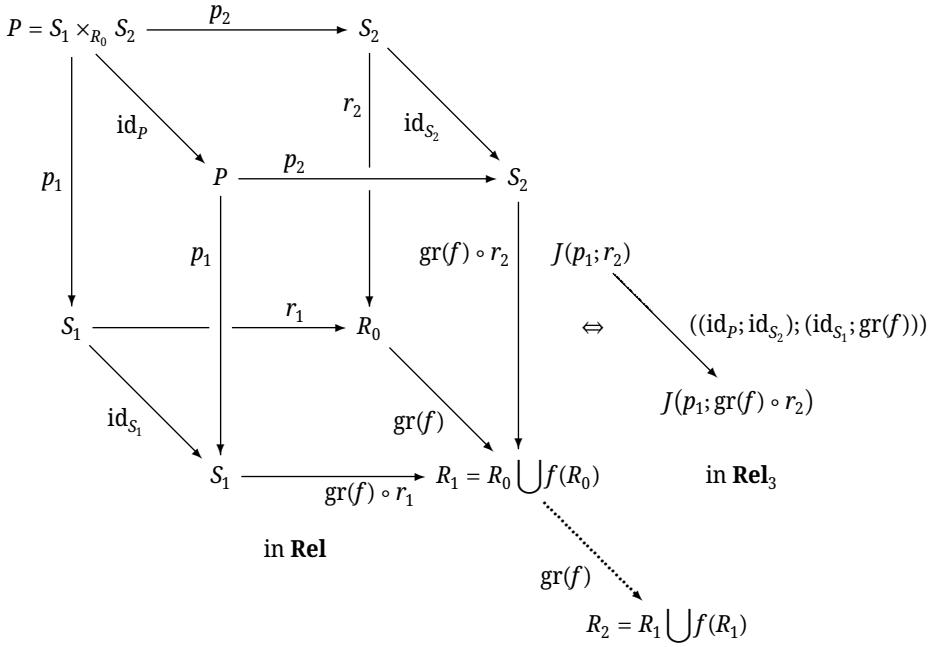
$$f : R_n \to R_{n+1}, \quad \text{where } R_{n+1} = R_n \bigcup f(R_n) \subseteq M_1 \times M_2. \tag{6.53}$$

It is easy to verify that, for two given trees, the function f is monotone. That is, for two different sets $R_0 \subseteq R_0'$ of bisimilar leafs in (6.49), for the image of f we obtain $f(R_0) \subseteq f(R_0')$, so that there exists a greatest fixed-point $\text{gfp}(f) = \bigcup\{R_k \mid R_k \subseteq f(R_k), k \geq 0\}$.

The propagation of computation of the strong bisimulation (greatest fixed point) from the leafs to the roots of the trees, can be shown by the following proposition.

Proposition 20. *Given a pullback diagram* (6.51) *with pullback object $P = S_1 \times_{R_0} S_2$ in* **Rel** *of the strongly bisimilar leafs of a finite abstract trees (prefix-closed set) $M_k \subseteq N^*$, for $k = 1,2$, the fixed point of the arrow $((\text{id}_P;\text{id}_{S_2});(\text{id}_{S_1};\text{gr}(f)))$ in $\mathbf{Rel}_3$, where $\text{gr}(f)$ is graph of the surjective function f given by* (6.52) *and* (6.53)*, defines the strong bisimulation.*

Proof. Let us consider the following commutative diagram in **Rel**:



(6.54)

where by coinduction[4] (closed backward, in the direction toward the tree's roots) in $\mathbf{Rel}_3$ n-times of the arrow $((\mathrm{id}_P; \mathrm{id}_{S_2}); (\mathrm{id}_{S_1}; \mathrm{gr}(f)))$ at some step $n \geq 1$ we obtain the greatest post-fixed-point[5] condition $R_{n+1} = R_n$, so that the greatest bisimulation relation is R is the union of all bisimulations R_k,

$$\begin{aligned} R &= \bigcup \{(f_{S_1}(x_i), f_{S_2}(x_j)) \mid (x_i, x_j) \in R_k \text{ is bisimulation for } 0 \leq k \leq n\} \\ &= \Big\{(f_{S_1}(x_i), f_{S_2}(x_j)) \mid (x_i, x_j) \in \mathrm{gfp}(f) = \bigcup \{R_k \mid R_k \text{ is bisimulation for } 0 \leq k \leq n\}\Big\}. \end{aligned} \tag{6.55}$$

4 The coinductive proof principle allows us to infer that a set is included in the coinductive set by proving that the given set satisfies the backward closure.

5 In 1980, Park was one of the top experts in fixed-point theory at the time. He makes the final step in the discovery of bisimulation precisely guided by fixed-point theory. Park notices that the inductive notion of equivalence that Milner is using for his CCS processes is based on a monotone functional over a complete lattice. And by adapting an example by Milner, he sees that Milner's equivalence is not a fixed point for the functional, and that therefore the functional is not cocontinuous. He then defines bisimilarity as the greatest fixed point of the functional, and derives the bisimulation proof method from the theory of greatest fixed points.

We use the following rule as a *coinductive* definition of bisimilarity ("equivalence") of states:

$$\frac{\forall a, s'.s \xrightarrow{a} s' \Rightarrow \exists t'.t \xrightarrow{a} t' \wedge (s' \approx_s t') \qquad \forall a, t'.t \xrightarrow{a} t' \Rightarrow \exists s'.s \xrightarrow{a} s' \wedge (s' \approx_s t')}{s \approx_s t}$$

This rule[6] is implemented by the definition of the function f in (6.52) and (6.53).

When transition systems are rooted, we can then say that two systems are bisimilar if their roots are. □

Bisimulation was first introduced in the context of modal logic by Van Benthem [50] to give an account of which subfamily of first-order logic is definable in modal logic.

Let us denote by $(\mathcal{M}, s)$ a pointed Kripke structure for a fixed state ("world") $s \in S$ in a Kripke model $\mathcal{M} = (S, \{R_a\}_{a \in A}, v)$. Let $\mathcal{C}$ be a class of pointed Kripke structures for a given multimodal propositional logic $\mathcal{L}$ and $\approx$ be an equivalence relation on $\mathcal{C}$.

Then $\mathcal{L}$ is preserved under $\approx$ over $\mathcal{C}$, or $\mathcal{L}$ is $\approx$-*invariant* over $\mathcal{C}$ iff for any $\mathcal{U} \approx \mathcal{U}'$ in $\mathcal{C}$ and any formula $\phi \in \mathcal{L}$, we have that $\mathcal{U} \models \phi$ iff $\mathcal{U}' \models \phi$, i. e., if $\mathcal{U}$ and $\mathcal{U}'$ are $\mathcal{L}$-equivalent. That is, the equivalence $\approx$ is a refinement of $\equiv_{\mathcal{L}}$, or $\approx \subseteq \equiv_{\mathcal{L}}$.

Invariance phenomena give insights into the semantics of the logic involved, and also often provide key tools for the model theoretic study of the logic (e. g., model constructions guided by $\approx$ equivalence). The relationship between modal logics and bisimulation equivalences provides an excellent example of such a fruitful companionship. The theorem of bisimulation invariance demonstrate that many-valued propositional logic $\mathcal{L}$ is *bisimulation invariant* when:

$$\text{if } (\mathcal{M}, s) \rightleftarrows (\mathcal{M}', s') \quad \text{(see Definition 33), then } (\mathcal{M}, s) \equiv_{\mathcal{L}} (\mathcal{M}', s').$$

Many features of the "modal character" can be understood in terms of *bisimulation invariance*; this is true most notably of the local and restricted nature of quantification. Due to these features, modal logic enjoys very specific features.

From the modal Lindström theorem (one notable example is Van Benthem's [50, 51] recent Lindström theorem for modal logic), an extension of basic modal logic satisfies compactness and *bisimulation invariance* if and only if it is no more expressive than basic modal logic. It is a classical result by van Benthem that modal logic is precisely the bisimulation invariant fragment of first-order logic. This means that an arbitrary first-order formula (in a vocabulary of unary and binary relations) is invariant under bisimulation if, and only if, it is equivalent to a formula of multimodal propositional logic.

6 This fixed-point computation of bisimulation by using coinduction has been used in my PhD thesis in 1995.

Transition systems are models for basic process calculi, such as ACP, CCS and CSP. Bisimulation equivalence is a congruence for all the operators of these calculi. By permitting more general operators, whose rules for transitions belong to a general format, bisimulation equivalence turns out to be the least congruence induced by language equivalence [52]. Models for richer process calculi capturing value passing, mobility, causality, time, probability and locations have been developed. The basic notion of bisimulation has been generalized, often in a variety of different ways, to cover these extra features. Modal mu-calculus, μM, introduced by Kozen [53], has the required extra expressive power.

Bisimulation equivalence is a very fine equivalence between states. An interesting line of enquiry is to reconsider classical results in automata and language theory, replacing language equivalence with bisimulation equivalence. These results concern definability, closure properties and decidability/undecidability. Bisimulation and coinduction are generally considered as one of the most important contributions of concurrency theory to computer science. In concurrency, the bisimulation equality, called bisimilarity, is the most studied form of behavioral equality for processes, and is widely used.

Bisimulation and, more generally, coinduction are employed today in a number of areas of computer science: functional languages, object-oriented languages, types, data types, domains, databases [5], compiler optimizations, program analysis, verification tools, etc. Also, the development of final semantics (Aczel [54, 55]; Rutten and Turi [56]; Jacobs and Rutten [57]), an area of mathematics based on coalgebras and category theory and used in the semantics of programming languages [5, 58–67], has been largely motivated by the interest in bisimulation. Final semantics also give us a rich and deep perspective on the meaning of coinduction and its duality with induction.

In the rest of this section, we will provide an interesting case for the possible application of the bisimulation.

6.6.1 Mathematics via symmetry: Reduction of many valued into 2-valued logic

In this section, we will consider the mathematical concept of symmetry for the transformation of many-valued modal (epistemic) into 2-valued modal logic programs by ontological encapsulation of predicates using *semantic reflection transformation*. It is interesting for an eventual future work of application of the fixed-point method for the bisimulation between these two modal logics invariant w. r. t. the knowledge representation.

The approach based on the contextualization of many-valued logics with the introduction of higher-order Herbrand interpretation types, introduces explicitly the coexistence of a set of algebraic truth values of original modal many-valued logic, transformed as parameters (or possible worlds) and the set of classic two logic values. This approach offers the possibility of using the standard semantics based on Herbrand interpretations.

Moreover, it uses the properties of the higher-order Herbrand types, as their fundamental nature is based on autoreferential Kripke semantics [68] where the possible worlds are algebraic truth values of original modal (epistemic) many-valued logic.

This autoreferential Kripke semantics, which has the possibility of flattening higher-order Herbrand interpretations into ordinary 2-valued Herbrand interpretations, gives us a clearer insight into the relationship between many-valued and 2-valued multimodal logics. Following this, we generalize the reduction to general structural many-valued logics, in an abstract way, based on Suszko's informal nonconstructive idea. In all cases, by using developed 2-valued reductions we obtain a kind of nontruth-valued modal metalogics, where two-valued formulae are modal sentences obtained by application of particular modal operators to original many-valued formulae.

In what follows, we will consider a general case of a many-valued logics based on bounded lattice $(X, 0, 1, \leq \ .\wedge, \vee)$ with antitonic negation "$\sim$" (such that hold De Morgan rules) top value 1 and bottom value 0. The reduction of many-valued logics into the standard 2-valued logic was considered by Suszko [69], where he illustrated how Lukasiewicz's 3-valued logic could be given a 2-valued, nontruth-functional, semantics. The main point, according to Suszko, is to make a distinction between the *algebraic truth values* in X of many-valued logics, which were supposed to play a merely referential role, while only two *logical truth values* in $\mathbf{2} = \{0, 1\}$ (0 for false and 1 for true value) would really exist. It is also based on the fact that the abstract logic is based on a *consequence relation* that is bivalent: given a set of logic formulae S, a formula ϕ can be inferred from S or not, i. e., the answer to the question "*if ϕ is inferred from S*" can only be "Yes" or "No."

This point of view for "logic values" is also considered correct by other authors, and it is also applied in our case of an ontological encapsulation of many-valued algebraic logic programs into 2-valued logic programs. Moreover, in a 2-valued reduction, for any propositional formula ϕ that has an "algebraic truth-value" α, we can consider a 2-valued meta-sentence "*the truth-value of ϕ is α.*"

Suszko's thesis for the reduction of every Tarskian (monotonic) n-valued logic into a 2-valued logic is based on this division of a set of logic values into a subset of designated and undesignated elements, but it is quite a nonconstructive result. In fact, he does not explain how he obtained a 2-valued semantics, or how such a procedure could be effectively applied.

In the paper by D.Batens [70], the author proposes a sort of binary print of the algebraic truth values for the 2-valued reduction, where each truth value is to be put into one-to-one correspondence with one element of a set of conveniently long "equivalent" sequences of 0's and 1's. This method is similar to what had been proposed by D. Scott, a decade before [71]. But this method is not universally applicable, and thus cannot be effectively used. Some other authors argued against Suszko's thesis [72] using examples of paraconsistent logic and Malinowski's inferential many-valuedness. But recently in [73], based on Suszko's observations on complementarity of designated and undesignated elements, a method was exhibited for the effective implementation of Suszko's reduction to

a subclass of finite-valued truth-functional logics, whose truth values satisfy the particular assumption of separability, where the "algebraic truth-values" can be individualized by means of the linguistic resources of the logic. What is important for the present work is that they show that a reduction of truth-functional many-valued logic into 2-valued logic will simply make it lose truth functionality: in fact, my transformation will result in *modal* logics.

Consequently, we will use *a constructive* approach to Suszko's method, and will exhibit a method for the effective implementation of 2-valued reduction *for all kinds* of many-valued logics. It avoids the necessity of dividing (in a problematic way based on subjective opinions) a set of algebraic truth values into designated and undesignated disjoint subsets in order to define the satisfaction relation (i. e., entailment), by using the valuations (model-theoretic semantics): the entailment $S \models \phi$ means that every model (valuation) of S is a model of ϕ.

Let us consider a modal epistemic logic program P_G composed by a set of predicate and function symbols, P, F, respectively, with the Herbrand base H, composed the set of all ground atoms over the Herbrand universe $\Gamma_U = \mathcal{D} \bigcup \Omega$, where $\mathcal{D}$ is an ordinary domain and Ω is an infinite enumerable set of marked null values (Skolem constants), $\Omega = \{\omega_0, \omega_1, \ldots.\}$. We denote the set of all terms by $\mathcal{T}_S$, and its subset of ground terms by $\mathcal{T}_0$.

For example, any rule in a many-valued logic program $A \leftarrow B_1, \ldots, B_n$ is *satisfied* if, for a given valuation v, the algebraic truth value of the head is greater than the value of the body, i. e., if $v(A) \geq v^*(B_1 \wedge \cdots \wedge B_n)$ for a given Herbrand interpretation $v : H \rightarrow X$ (here, v^* extends the valuation v to all formulae).

Example 20. Let us consider the example $X = \mathcal{B}_4$ of Belnap's 4-valued epistemic logic program P_G. The truth and knowledge ordering relations and operations in a bilattice $\mathcal{B}_4$ propagates to the function space $\mathcal{B}_4^H$ of all Herbrand interpretations $v : H \rightarrow \mathcal{B}_4$. This makes a function space $\mathcal{B}_4^H$ itself a complete infinitary distributive bilattice [74].

Belnap's 4-valued bilattice [75], $X = \mathcal{B}_4 = \{t, f, \bot, \top\} = \{1, 0, \bot, \top\}$ is shown in Figure 6.1, where $t = 1$ is *true*, $f = 0$ is *false*, $\top$ is inconsistent (both true and false) or *possible*, and $\bot$ is *unknown*. In what follows, we denote by $x \bowtie y$ two unrelated elements in X (so that not ($x \leq y$ or $y \leq x$)).

As Belnap observed, these values can be given two natural orders: *truth* order, $\leq_t$, and *knowledge* order, $\leq_k$, such that $f \leq_t \top \leq_t t, f \leq_t \bot \leq_t t$, $\bot \bowtie_t \top$ and $\bot \leq_k f \leq_k \top$, $\bot \leq_k t \leq_k \top, f \bowtie_k t$. That is, bottom element 0 for $\leq_t$ ordering is f, and for $\leq_k$ ordering is $\bot$, and top element 1 for $\leq_t$ ordering is t, and for $\leq_k$ ordering is $\top$. Meet and join operators under $\leq_t$ are denoted $\wedge$ and $\vee$; they are natural generalizations of the usual conjunction and disjunction notions. Meet and join under $\leq_k$ are denoted $\otimes$ and $\oplus$, such that the following hold: $f \otimes t = \bot, f \oplus t = \top, \top \wedge \bot = f$ and $\top \vee \bot = t$.

There is a natural notion of the *bilattice truth negation*, denoted $\neg$, (reverses the $\leq_t$ ordering, while preserving the $\leq_k$ ordering): switching f and t, leaving $\bot$ and $\top$ and corresponding knowledge negation (*conflation*), denoted $-$, (reverses the $\leq_k$ ordering,

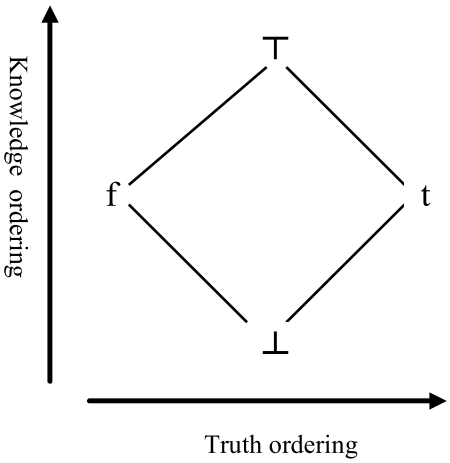


Figure 6.1: Belnap's bilattice.

while preserving the $\leq_t$ ordering), switching $\bot$ and $\top$, leaving f and t. These two kinds of negation commute: $-\neg x = \neg - x$ for every member x of a bilattice.

The conflation is a monotone function that preserves all finite meets (and joins) w. r. t. the lattice $(\mathcal{B}_4, \leq_t)$, thus it is the universal (and existential, because $- = \neg - \neg$) modal many-valued operator: "it is believed that" for a bilattice (as in ordinary 2-valued logic, the epistemic negation is composition of strong negation $\neg_t$ and this belief operator, $\neg = \neg_t -$), which extends the 2-valued belief of the autoepistemic logic as follows:

1. if A is true, then "it is believed that A," i. e., $-A$, is true;
2. if A is false, then "it is believed that A" is false;
3. if A is unknown, then "it is believed that A" is inconsistent: it is really inconsistent to believe in something that is unknown;
4. if A is inconsistent (i. e., *both* true and false), then "it is believed that A" is unknown: really, we cannot tell anything about believing in something that is inconsistent.

This belief modal operator can be used to define the *epistemic* (bilattice) negation $\neg$, as a composition of strong negation $\neg_t$ and this belief operator, i. e., $\neg = \neg_t -$. That is, the epistemic negation is negation as a modal operator [76] and as a paraconsistent negation [77, 78].[7] Such a *modal autoepistemic logic*, based on Belnap's bilattice, $(\mathcal{B}_4, \leq_t, \wedge, \vee, \neg_t, \rightharpoonup, -, \mu, f, t)$ can be used for logic programming with incomplete and partially inconsistent information [80]: in such a logic, we use only the epistemic negation $\neg = \neg_t -$.

7 We can introduce also Moore's *autoepistemic* operator [79], $\mu : \mathcal{B} \to \mathcal{B}$, for a Belnap's bilattice, defined by $\mu(x) = t$ if $x \in \{\top, t\}$; f otherwise. It is easy to verify that it is monotone w. r. t. the $\leq_t$, i. e., multiplicative ($\mu(x \wedge y) = \mu(x) \wedge \mu(u)$ and $\mu(t) = t$) and additive ($\mu(x \vee y) = \mu(x) \vee \mu(u)$ and $\mu(f) = f$); consequently, it is also a self-adjoint (contemporary universal and existential) modal operator, $\mu = \neg_t \mu \neg_t$. Notice that different from the belief modal operator $-$, Moore's modal operator μ is not surjective.

One of the key insights behind bilattices [79, 81] was the interplay between the truth values assigned to sentences and the (nonclassic) notion of *implication*. The problem was to study how truth values should be propagated "across" implications. In [82], it is proposed that the following IB based approach to the "object" 4-valued logic programming, which extends the definition, is given for a 3-valued logic programming [83]: Let P_B be the subset of built-in predicates. The valuation, $v : H \to \mathcal{B}_4$, is extended to the logic implication of a ground clause $p(\mathbf{c}) \leftarrow \phi$, where $\phi = B_1 \wedge \cdots \wedge B_n$, with $B_i \in H$, for $1 \leq i \leq n$, by

$$v^*(\phi \to p_j^n(\mathbf{c})) = t \quad \text{iff} \quad v(p_j^n(\mathbf{c})) \geq_t v^*(\phi) \text{ or } (p_j^n \in P_B \text{ and } v(\phi) = \top).$$

We assumed the *inconsistency acceptance* principle: if $p_j^n \in P_B$ is a built-in predicate, this clause is satisfied also when $v(p_j^n(\mathbf{c})) = f$ and $v^*(\phi) = \top$.

The *built-in predicates* (ex, $=, \leq, \geq, \ldots$) may be used for integrity constraints: let $p(x,y)$ be a predicate and we define the key constraint for attributes in x by $(y = z) \leftarrow p(x,y), p(x,z)$, where the atom $y = z$ is based on the built-in predicate $'=\!'$. Let us consider a program:

$$p(x,y) \leftarrow r(x,y), \quad (y = z) \leftarrow p(x,y),\ p(x,z),$$

where r is a source database relation with two tuples, $(a,b), (a,c)$, p is a virtual relation of this database with key constraint, and x, y, z are variables. The built-in predicates have the same prefixed extension in *each* model of a logic program, and their ground atoms are *true or false*. If we assume that, $r(a,b), r(a,c)$ are true, then such facts are mutually inconsistent for p because of key constraint ($b = c$ is false). Thus, only one of them may be true in any model of this logic program, e. g., $r(a,b)$. So, if we assign the "possible" value $\top$ to $r(a,c)$ (or to both of them), we obtain that the clause $(b = c) \leftarrow p(a,b),\ p(a,c)$, and thanks to the *inconsistency acceptance*, is satisfied.

The many-valued definition for implication, which generalizes the 2-valued definition of clause satisfaction for Lukasiewicz's and Kleene's strong 3-valued matrices [84], $f_{\leftarrow} : \mathcal{B}_4 \times \mathcal{B}_4 \to \mathcal{B}_4$, can be defined (with $\alpha = t$ and $\alpha = \bot$ for Lukasiewicz's and Kleene's case, respectively) by

$\rightarrow$	$\boldsymbol{t}$	$\bot$	$\top$	$\boldsymbol{f}$
t	t	$\bot$	$\top$	f
$\bot$	t	α	$\top$	$\bot$
$\top$	t	t	t	t
f	t	t	t	t

For our purpose, we assume the Lukasiewicz's extension, i. e., $\alpha = t$, in order to have a tautology $a \leftarrow a$ for any formula a, and also to guarantee the truth of a clause (implica-

tion) $p(\mathbf{c}) \leftarrow \phi$, whenever $v(p(\mathbf{c})) \geq_t v^*(\phi)$, as used in fixed-point semantics for "immediate consequence operators." Such conservative extensions are based on the following observation: the problem to study how the truth values should be propagated "across" implications can be restricted only to *true* implications (in fact, we do not use implications when they are not true, because the "immediate consequence operator" derives new facts only for *true* clauses, i. e., when implications used in the clauses are true).

I considered [85] two approaches for *predicate* many-valued logics (the propositional version can be considered as a special case, when all predicate symbols have a zero arity): the first one introduces the *unary* modal operator for each truth value of original many-valued logic; the second approach introduces the *binary* modal operator for each binary truth-valued logic operator (conjunction, disjunction, implication) of original many-valued logic. Both of them transform an original truth-functional many-valued logic into *nontruth-functional* 2-valued modal logic, and here we will consider only the second one, based on the ontological embedding into the syntax of new 2-valued encapsulated many-valued logic.

The "object" many-valued logic is based on the lattice X of truth values, which are *epistemic*. Sentences are to be marked with some of these lattice logic values, according as to what the computer has been told—or with only a slight metaphor, according to what it *believes or knows*. Of course, these sentences *have* also Frege's ontological truth values (true and false), independently of what the computer has been told: we want that the computer can use also these ontological "meta" knowledge. For example, the computer believes that the sentence ϕ has a value $a \in X$; then the "meta" sentence, *"I (computer) believe that ϕ has a value a"* is *ontologically true*. The many-valued encapsulation is just the way to pass from the epistemic ("object") many-valued logic into ontological ("meta") 2-valued logic.

It can be seen as a flattening of a many-valued logic, where an algebraic truth-value $\beta \in X$ of an original ground atom $p_j^k(c_1, \ldots, c_k)$ is deposited into the logic attribute of a new predicate r_F, obtained by an extension of the old predicate p_j^k, so that we obtain the "flattened" 2-valued ground atom $r_F(c_1, \ldots, c_k, \beta)$. In that case, we will obtain the *positive* multimodal logic programs with binary modal operators for conjunction, disjunction and implication and the unary modal operator for negation.

6.6.2 Many-valued knowledge invariance through modal logic transformations: Semantic reflection

The flattening of an original many-valued lattice-based program into a modal metalogic is a kind of *ontological-encapsulation*, where the encapsulation of an original many-valued logic program into the 2-valued modal metalogic program corresponds to this flattening process. This approach is developed in a number of papers, and more information can be found in [7, 62, 84, 86, 87]. Here, we will present a slightly modified

version. Such an encapsulation is characterized by having capability for *semantic reflection*: intuitively, for each predicate symbol we need some function, which *reflects* its logic semantic over a domain Γ_U. We introduce also the set of functional symbols κ_p over a domain Γ_U in our logical language in order to obtain an enriched logical language [84] where we can encapsulate the "object" (ordinary) many-valued logic programming. Such set of functional symbols will be derived from the lattice-semantic mapping $\mathcal{K}$. We will also introduce a new symbol **e** (for "error condition"), necessary in order to render *complete* the functions for a generalized interpretation and a semantic reflection.

Definition 35. Let P_G be a many-valued logic program with a set of predicate and functional symbols P and F, respectively, with a Herbrand model $v : H \to X$ where H is a Herbrand base, with a set $\mathcal{T}_0$ of all ground terms and a set $\mathcal{T} = \bigcup_{k\in\mathcal{N}} \mathcal{T}_0^k$ with $\mathcal{N} = \{1, 2, \ldots, n\}$ where n is the maximal arity of symbols in $P \bigcup F$.

1. A *generalized interpretation* is a mapping $\mathcal{I} : P \times \mathcal{T} \to X \bigcup \{\mathbf{e}\}$, such that for any $\mathbf{c} = (c_1, \ldots, c_n) \in \mathcal{T}$, $\mathcal{I}(p, \mathbf{c}) = v(p(\mathbf{c}))$ if $\mathrm{ar}(p) = n$; **e** otherwise.
2. A *semantic reflection* is defined by a mapping $\mathcal{K} = \Lambda(\mathcal{I}) : P \to (X \bigcup \{\mathbf{e}\})^{\mathcal{T}}$, where Λ is the currying operator from lambda calculus used in diagram A.19 in Section A.2. For each $p \in P$ that is not a built-in 2-valued predicate, we define a new functional symbol κ_p for a mapping $\mathcal{K}(p) : \mathcal{T} \to X \bigcup \{\mathbf{e}\}$.
3. If p is a 2-valued built-in predicate, then the mapping κ_p is defined uniquely and independently of v, by: for any $\mathbf{c} \in \mathcal{T}_0^{\mathrm{ar}(p)}$, $\kappa_p(\mathbf{c}) = 1$ if $p(\mathbf{c})$ is true; 0 otherwise.

We recall the well-known fact that 2-valued *built-in* predicates (as $\leq$, $=$, etc.) have constant extensions in any Herbrand interpretation (they preserve *the same meaning* for any logic interpretation, differently from ordinary predicates).

A semantic-reflection $\mathcal{K}$, obtained from a generalized interpretation $\mathcal{I}$, introduces a function symbol $\kappa_p = \mathcal{K}(p)$ for each predicate $p \in P$ of the original logic program P_G, such that for any $\mathbf{c} = (c_1, \ldots, c_n) \in \mathcal{T}$, it holds that $\kappa_p(\mathbf{c}) = v(p(\mathbf{c}))$ if $\mathrm{ar}(p) = n$; $\{\mathbf{e}\}$ otherwise. These new function symbols will be used in a new metalogic language, used to transform each original many-valued atom p in P into a new atom p_F obtained as an extension of the original atom p by one "logic" attribute with the domain of values in X. The interpretation of κ_p in this new metalogic program has to reflect the meaning of the original many-valued predicate p in the original many-valued logic program P_G.

Consequently, we are able to introduce a program encapsulation (flattening) transformation $\mathcal{E}$. The many-valued ground atoms of a many-valued logical language $\mathcal{L}_{\mathrm{mv}}$ can be transformed in "encapsulated" atoms of a 2-valued logic in the following simple way [84]: the original (many-valued) fact that the ground atom $A = p_j^n(c_1, \ldots, c_n)$, of the n-ary predicate $p_j^n \in P$, has an epistemic value $\alpha = \kappa_p(c_1, \ldots, c_n)$ in $\mathcal{B}_4$. We transform in encapsulated atom $p_F(c_1, \ldots, c_n, \alpha)$ with meaning "*A has a value α.*" Indeed, what we do is to *replace* the original n-ary predicate $p_j^n(x_1, \ldots, x_n)$ with an $n + 1$-ary predicate $p_F(x_1, \ldots, x_n, \alpha)$ (flattening), with the added logic-attribute α. It is easy to verify that for

any given many-valued valuation $v \in X^H$, each ground atom $p_F(c_1, \ldots, c_n, \alpha)$ is ontologically true (when $\alpha = v(p_j^n(c_1, \ldots, c_n))$) or false. Let EMV denote this new 2-valued encapsulation of many-valued logic *for logic programming*.

Syntax

We distinguish between what the reasoner believes in (at the *object* (epistemic many-valued sublanguage) level), and what is actually true or false in the real world (at the EMV ontological "meta" level); thus, roughly, the "meta" level is an (classic) encapsulation of the object level.

Definition 36. Let P_G be an "object" many-valued logic program with the set of predicate symbols P. The translation in the syntax of encapsulated program P_F is as follows:

1. Each positive literal $p_j^n \in P$, $\mathcal{E}(p_j^n(x_1, \ldots, x_n)) = p_F(x_1, \ldots, x_n, \kappa_p(x_1, \ldots, x_n))$;
   Each negative literal in P, $\mathcal{E}(\sim p_j^n(x_1, \ldots, x_n)) = p_F(x_1, \ldots, x_n, \sim \kappa_p(x_1, \ldots, x_n))$; and we denote by P_E the set of new obtained predicates p_F;
2. $\mathcal{E}(\sim \phi) = \sim_F \mathcal{E}(\phi)$;
3. $\mathcal{E}(\phi \odot \varphi) = \mathcal{E}(\phi) \odot_F \mathcal{E}(\varphi)$; for $\odot \in \{\wedge, \vee, \leftarrow\}$, where $\sim_F, \wedge_F, \vee_F$ and $\leftarrow_F$ are new syntax symbols for the negation, conjunction, disjunction and implication at the encapsulated 2-valued "meta" level.

Thus, the obtained "meta" program is equal to $P_F = \{\mathcal{E}(\phi) \mid \phi \text{ is a clause in } P_G\}$, with the 2-valued Herbrand base $H_F = \{p_F(c_1, \ldots, c_n, \alpha) \mid p_j^n(c_1, \ldots, c_n) \in H \text{ and } \alpha \in X\}$.

We denote by $\mathcal{L}_F$ the set of formulae obtained from the set of predicate letters in P_E and new operators $\sim_F, \wedge_F, \vee_F$ and $\leftarrow_F$.

This embedding of the many-valued "object" logic program P_G into a 2-valued "meta" logic program P_F is an *ontological* embedding [84]: we consider the formulae of P_G as beliefs and interpret negation $\sim p_j^n(x_1, \ldots, x_n)$ in rather restricted sense–as a belief in the falsehood of $p_j^n(x_1, \ldots, x_n)$, rather as not believing that $p_j^n(x_1, \ldots, x_n)$ is true (like in an ontological embedding for classical negation). Like for Moore's autoepistemic operator, for the encapsulation modal operator $\mathcal{E}$, $\mathcal{E}\phi$ is intended to capture the notion of, "*I know that ϕ has a value $v(\phi)$,*" for a given valuation v of the "object" logic program.

Let $\mathcal{L}_{\text{mv}}^G$ be the set of all sentences defined by this Herbrand base H and lattice operations (included a many-valued implication $\leftarrow$ also), with $X \subseteq \mathcal{L}_{\text{mv}}^G$. So, we have the following.

Corollary 15. *The encapsulation operator $\mathcal{E}$ is:*

1. *Modal operator, such that, for any $\alpha \in X$, $\mathcal{E}(\alpha) = t$ if $\alpha = t$; f otherwise.*
2. *Homomorphism between the "object" algebra $(\mathcal{L}_{\text{mv}}, \wedge, \vee, \leftarrow)$ with a carrier set of (positive and negative) literals, and "meta" algebra $(\mathcal{L}_F, \wedge_F, \vee_F, \leftarrow_F)$, where $\wedge_F, \vee_F$ and $\leftarrow_F$ are 2-valued operators corresponding to lattice's meet and join, respectively (denoted also by $\wedge, \vee$).*

Semantics

With the transformation of the original "object" logic program P_G into its annotated "meta" version program P_F, we obtain an *always positive* consistent logic program.

A Herbrand interpretation of P_F is a 2-valued mapping $I_F : H_F \to \mathbf{2}$. We denote by $\mathbf{2}^{H_F}$ the set of all interpretations from H_F into $\mathbf{2}$, and by X^H the set of all Herbrand many-valued interpretations, from H to the lattice X. The meaning of the *encapsulation* of this "object" logic program P_G into this "meta" logic program P_F is fixed into the kind of interpretation to give to such new introduced functional symbols $\kappa_p = \mathcal{K}(p)$: in fact, *we want* [82] that they reflect the semantics of the "object" level logic program P_G.

Definition 37 (Satisfaction). The encapsulation of an epistemic "object" logic program P_G into the "meta" program P_F means that, for any many-valued Herbrand interpretation $v \in X^H$, the function $\kappa_p = \mathcal{K}(p)$, $p \in P$ reflects this semantics, i. e., for any tuple $\mathbf{c} \in \mathcal{T}_0^{\text{arity}(p)}$,

$$\kappa_p(\mathbf{c}) = v(p(\mathbf{c})).$$

So, we obtain a mapping, $\Theta : X^H \to \mathbf{2}^{H_F}$, such that $I_F = \Theta(v) \in \mathbf{2}^{H_F}$ with: for any ground atom $p(\mathbf{c})$, $I_F(\mathcal{E}(p(\mathbf{c}))) = t$, if $\kappa_p(\mathbf{c}) = v(p(\mathbf{c}))$; f otherwise.

Let $g : \mathcal{V} \to \Gamma_U$ be an assignment to variables in $\mathcal{V}$ such that $g(\mathcal{E}(p_j^n(x_1, \ldots, x_n))) = \mathcal{E}(p_j^n(g(x_1), \ldots, g(x_n)))$, and

1. $I_F \vDash_g \mathcal{E}(p_j^n(x_1, \ldots, x_n))$ iff $\kappa_p((g(x_1), \ldots, g(x_n))) = v(p_j^n(g(x_1), \ldots, g(x_n)))$.
2. $I_F \vDash_g \mathcal{E}(\sim p_j^n(x_1, \ldots, x_n))$ iff $\kappa_p((g(x_1), \ldots, g(x_n))) = \sim v(p_j^n(g(x_1), \ldots, g(x_n)))$.

Notice that in this semantics the "meta" implication used in logic program clauses $\leftarrow_F$ between two sentences ϕ and ψ, in $\mathcal{E}(\phi) \leftarrow_F \mathcal{E}(\psi) = \mathcal{E}(\phi \leftarrow \psi)$, is based on the "object" epistemic many-valued implication $\leftarrow$ (which *is not classical*, i. e., $\phi \leftarrow \psi \neq \phi \vee \sim \psi$) and determines how the logical value of a body of clause "propagates" to its head. That is, $\mathcal{E}(\phi) \leftarrow_F \mathcal{E}(\psi)$ is true if $v(\phi/g \leftarrow \psi/g)$ is true. So, the semantics of the logic operators in this "meta" (encapsuled) level are modal operators.

Corollary 16. *The semantics of encapsulation $\mathcal{E}$ is obtained by identifying the semantic reflection with the λ-abstraction of the generalized Herbrand interpretation, $\mathcal{K} = \Lambda(\mathcal{I})$, so that the semantics of many-valued logic programs can be determined by $\mathcal{I}$ (at "object" level) or, equivalently, by its reflection $\mathcal{K}$ (at encapsulated or "meta" level).*

Proof. From $\mathcal{K} = \Lambda(\mathcal{I})$, where Λ is the currying operator from lambda calculus used in diagram A.19 in Section A.2, we obtain that for any $p(\mathbf{c}) \in H$ holds $v(p(\mathbf{c})) = \mathcal{I}(p, \mathbf{c}) = \Lambda(\mathcal{I})(p)(\mathbf{c}) = \mathcal{K}(p)(\mathbf{c}) = \kappa_p(\mathbf{c})$. What is the semantic of encapsulation? □

We can consider the λ-abstraction of the generalized Herbrand interpretation as epistemic semantics, because given a Herbrand many-valued (epistemic) interpretation $v : H \to X$, for any predicate p and constant $\mathbf{c} \in \mathcal{T}_0^{\text{arity}(p)}$, it holds that $\Lambda(\mathcal{I})(p)(\mathbf{c}) =$

$v(p(\mathbf{c}))$. So, the semantic of encapsulation may be expressed by the following knowledge invariance:

"ontological semantic-reflection ≡ epistemic semantics," i. e., $\mathcal{K} = \Lambda(\mathcal{I})$

These semantics have been used to provide a coalgebraic semantics for logic programs [60].

Remark. The new introduced logic symbols $\sim_F, \wedge_F, \vee_F$ and $\leftarrow_F$ for the metalogic connectives are not truth-functional as are original many-valued operators but modal. Hence, by this transformation of P_R we obtain a modal logic program P_F that is a *positive* logic program (without negation). Different from a sentence $\phi \in \mathcal{L}^G_{\text{mv}}$, the transformed metaformula $\Phi = \mathcal{E}(\phi) \in \mathcal{L}_F$ can be only *true or false* in a given possible world $w \in \mathcal{W} = X$ for this metamodal logic (in a given Kripke model $\mathcal{M}$ of obtained metalogic program P_F).

In this definition of a metalogic program P_F, the set of mappings $\{\kappa_p = \mathcal{K}(p) \mid p \in P\}$ is considered as a set of *built-in functions*, determined by a given semantic reflection $\mathcal{K}$, that extends a given set of functional symbols in F. □

The encapsulation operator $\mathcal{E}$ is intended to have the following property for a valuation $v^* : \mathcal{L}^G_{\text{mv}} \to X$ (a homomorphic extension of Herbrand interpretation v to all formulae in $\mathcal{L}^G_{\text{mv}}$) of a many-valued logic program P_G: for any sentence ϕ, the encapsulated metaformula $\mathcal{E}(\phi)$ intends to capture the notion of ϕ with its value $v(\phi)$ as well, in the way that

"$\mathcal{E}(\phi)$ *is true exactly in the possible world* $w = v^*(\phi) \in \mathcal{W} = X$."

In order to introduce a concept of *absolute truth or falsity* (not relative to a single possible world in $\mathcal{W} = X$) for the sentences in $\mathcal{L}_F$, we need a new autoepistemic modal operator $\Diamond$. Consequently, for any given sentence $\Phi \in \mathcal{L}_F$, similar to Moore's autoepistemic operator, a formula $\Diamond\Phi$ is able to capture the 2-valued notion of

"Φ *is a semantic reflection of a many-valued logic program model* v."

Notice that in this encapsulation, e. g., the metaimplication $\leftarrow_F$ derived from the many-valued implication, $\mathcal{E}(\phi) \leftarrow_F \mathcal{E}(\psi) = \mathcal{E}(\phi \leftarrow \psi)$, specifies how, for a given clause in P_G, a logic value of the body "propagates" to the head of this clause. It is not functionally dependent on the truth values of its arguments, thus it must be a binary *modal* operator. The idea to use ternary relations to model binary modal operators comes from relevance logic [88–91]. So, based on the autoreferential semantics (where the set of possible worlds $\mathcal{W}$ is taken to be just the set of logical values X of the many-valued logic), we can define the following Kripke models.

Definition 38. Let P_G be a many-valued logic program with a set of predicate symbols P, a many-valued Herbrand model $v : H \to X$ and its semantic-reflection $\mathcal{K}$.

Then the model of the flattened program P_F in Definition 36 is defined as the Kripke-style model $\mathcal{M} = (\mathcal{W}, \{\mathcal{R}_\sim, \mathcal{R}_\wedge, \mathcal{R}_\vee, \mathcal{R}_\rightarrow, \mathcal{R}_\times = \mathcal{W} \times \mathcal{W}\}, \mathcal{D}, I_K)$ with $\mathcal{W} = X$, where

$$\mathcal{R}_\wedge = \{(x \wedge y, x, y) \mid x, y \in \mathcal{W}\}, \quad \mathcal{R}_\vee = \{(x \vee y, x, y) \mid x, y \in \mathcal{W}\},$$
$$\mathcal{R}_\rightarrow = \{(x \rightarrow y, x, y) \mid x, y \in \mathcal{W} \text{ and } x \leq y\}, \quad \mathcal{R}_\sim = \{(\sim x, x) \mid x \in \mathcal{W}\},$$

and $I_K : \mathcal{W} \times P_E \rightarrow \bigcup_{n \in \mathcal{N}} \mathbf{2}^{\mathcal{D}^n \times \mathcal{W}}$, such that for any $p_j^n \in P$ with arity n (i. e., $p_F \in P_F$ with arity $n+1$), a tuple of constants $(c_1, \ldots, c_n) \in \mathcal{D}^n$ and a world $w \in \mathcal{W}$,

$$I_K(w, p_F)(c_1, \ldots, c_n, \alpha) = 1 \quad \text{iff} \quad w = \alpha = \kappa_p(c_1, \ldots, c_n),$$

such that, for any formula $\Phi, \Psi \in \mathcal{L}_F$, the satisfaction relation $\vDash_{w,g}$, for a given assignment g and a world $w \in \mathcal{W}$, is defined as follows:

1. $\mathcal{M} \vDash_{w,g} p_F(x_1, \ldots, x_n, \alpha)$ iff $I_K(w, p_F)(g(x_1), \ldots, g(x_n), \alpha) = 1$.
2. $\mathcal{M} \vDash_{w,g} \sim_F \Phi$ iff $\exists y((w, y) \in \mathcal{R}_\sim$ and $\mathcal{M} \vDash_{y,g} \Phi)$.
3. $\mathcal{M} \vDash_{w,g} \wedge_F(\Phi, \Psi)$ iff $\exists y, z((w, y, z) \in \mathcal{R}_\wedge$ and $\mathcal{M} \vDash_{z,g} \Phi$ and $\mathcal{M} \vDash_{y,g} \Psi)$.
4. $\mathcal{M} \vDash_{w,g} \vee_F(\Phi, \Psi)$ iff $\exists y, z((w, y, z) \in \mathcal{R}_\vee$ and $\mathcal{M} \vDash_{z,g} \Phi$ and $\mathcal{M} \vDash_{y,g} \Psi)$.
5. $\mathcal{M} \vDash_{w,g} \leftarrow_F (\Phi, \Psi)$ iff $\exists y, z((w, y, z) \in \mathcal{R}_\rightarrow$ and $\mathcal{M} \vDash_{z,g} \Phi$ and $\mathcal{M} \vDash_{y,g} \Psi)$.
6. $\mathcal{M} \vDash_{w,g} \Diamond\Phi$ iff $\exists y((w, y) \in \mathcal{R}_\times$ and $\mathcal{M} \vDash_{y,g} \Phi)$.

The binary operators $\wedge_F, \vee_F$ and $\leftarrow_F$ for this multimodal logic are the existential modal operators w. r. t. the ternary relation $\mathcal{R}_\wedge$, $\mathcal{R}_\vee$ and $\mathcal{R}_\rightarrow$, respectively, while $\sim_F$ and $\Diamond$ are the existential unary modal operator w. r. t. the binary relation $\mathcal{R}_\sim$ and $\mathcal{R}_\times$, respectively.

Proposition 21. *For any assignment g and a formula $\Phi \in \mathcal{L}_F$, we have that $\|\Diamond\Phi/g\| \in \{\emptyset, \mathcal{W}\}$, where $\emptyset$ is the empty set and $\mathcal{W} = X$ is the set of truth values. That is, for any many-valued formula $\phi \in \mathcal{L}_{\mathrm{mv}}$ the formula $\Diamond\mathcal{E}(\phi/g)$ is true in the Kripke-style relational model $\mathcal{M}$ given by Definition* 38*, so that a Kripke-style model $\mathcal{M}$ of P_F corresponds to the many-valued algebraic model v of the original program P_G.*

Proof. In what follows, we denote by v^* the (homomorphic) extension of a Herbrand model v to all sentences. Let us demonstrate that for any $\phi \in \mathcal{L}$, i. e., $\mathcal{E}(\phi) \in \mathcal{L}_F$, holds that $\mathcal{M} \vDash_{w,g} \mathcal{E}(\phi)$ iff $w = v^*(\phi/g)$.

1. For any atomic formula $p(x_1, \ldots, x_n)$, we have that $\mathcal{M} \vDash_{w,g} \mathcal{E}(p(x_1, \ldots, x_n))$ iff $V(w, p_F)(g(x_1), \ldots, g(x_n), \kappa_p(g(x_1), \ldots, g(x_n))) = 1$ iff $w = \kappa_p(g(x_1), \ldots, g(x_n)) = \Lambda(\mathcal{I})(p)(g(x_1), \ldots, g(x_n)) = v(p(g(x_1), \ldots, g(x_n))) = v(p(x_1, \ldots, x_n)/g)$. Vice versa, if $w = v(p(x_1, \ldots, x_n)/g)$, i. e., $w = v(p(x_1, \ldots, x_n)/g) = \kappa_p(g(x_1), \ldots, g(x_n))$, then $V(w, p_F)(g(x_1), \ldots, g(x_n), \kappa_p(g(x_1), \ldots, g(x_n))) = 1$ and, consequently, from point 1 of definition above, $\mathcal{M} \vDash_{w,g} \mathcal{E}(p(x_1, \ldots, x_n))$.

Suppose, by the inductive hypothesis, that $\mathcal{M} \vDash_{z,g} \mathcal{E}(\phi)$ iff $z = v^*(\phi/g)$, and $\mathcal{M} \vDash_{y,g} \mathcal{E}(\psi)$ iff $y = v^*(\psi/g)$, then:

2. For any formula $\varphi =\sim \phi$, we have that $\mathcal{M} \vDash_{w,g} \mathcal{E}(\varphi)$ iff $\mathcal{M} \vDash_{w,g} \mathcal{E}(\sim \phi)$ iff $\mathcal{M} \vDash_{w,g\sim_F} \mathcal{E}(\phi)$ iff $(\exists z((w,z) \in \mathcal{R}_\sim$ and $\mathcal{M} \vDash_{z,g} \mathcal{E}(\phi)))$, i. e., if

$$\begin{aligned} w &=\sim z \quad \text{(from the definition of accessibility relation } \mathcal{R}_\sim) \\ &=\sim v^*(\phi/g) = v^*(\sim \phi/g) \quad \text{(from a homomorphic property of } v^*) \\ &= v^*(\varphi/g). \end{aligned}$$

Vice versa, if $w = v^*(\varphi/g) = v^*(\sim \phi)/g =\sim v^*(\phi/g) =\sim z$ then, from the inductive hypothesis, $\mathcal{M} \vDash_{w,g\sim_F} \mathcal{E}(\phi)$, i. e., $\mathcal{M} \vDash_{w,g} \mathcal{E}(\varphi)$.

3. For any formula $\varphi = \phi \odot \psi$, where $\odot \in \{\wedge, \vee, \rightarrow\}$, we have that $\mathcal{M} \vDash_{w,g} \mathcal{E}(\varphi)$ iff $\mathcal{M} \vDash_{w,g} \mathcal{E}(\phi \odot \psi)$ iff $\mathcal{M} \vDash_{w,g} \mathcal{E}(\phi) \odot_F \mathcal{E}(\psi)$ iff $(\exists y, z((w,y,z) \in \mathcal{R}_\odot$ and $\mathcal{M} \vDash_{z,g} \mathcal{E}(\phi)$ and $\mathcal{M} \vDash_{y,g} \mathcal{E}(\psi)))$, i. e., if

$$\begin{aligned} w &= z \odot y \quad \text{(from a definition of relation } \mathcal{R}_\odot) \\ &= v^*(\phi/g) \odot v^*(\psi/g) = v^*(\phi/g \odot \psi/g) \quad \text{(from a homomorphic property of } v^*) \\ &= v^*\big((\phi \odot \psi)/g\big) = v^*(\varphi/g). \end{aligned}$$

Vice versa, if $w = v^*(\varphi/g) = v^*(\phi \odot \psi)/g = v^*(\phi/g) \odot v^*(\psi/g) = z \odot y$ then, from the inductive hypothesis, $\mathcal{M} \vDash_{w,g} \mathcal{E}(\phi) \wedge_F \mathcal{E}(\psi)$, i. e., $\mathcal{M} \vDash_{w,g} \mathcal{E}(\varphi)$.

Thus, for any $\Phi \in \mathcal{L}_F$ we have that $\|\Phi/g\| = \{w\}$ for some $w \in \mathcal{W}$, if $\Phi/g = \mathcal{E}(\phi/g)$; otherwise $\|\Phi/g\| = \emptyset$. Consequently, we have that $\|\Diamond\Phi/g\| = \{w \mid \exists y((w,y) \in \mathcal{R}_\times$ and $\mathcal{M} \vDash_{y,g} \Phi)\} = \mathcal{W}$ if $\Phi/g = \mathcal{E}(\phi/g)$; otherwise $\|\Diamond\Phi/g\| = \emptyset$. That is, each sentence $\Diamond\Phi/g$ for any $\Phi \in \mathcal{L}_F$ is a *2-valued formula*.

From Definition 38, we have seen how a many-valued model v of a logic program P_G uniquely determines a Kripke model $\mathcal{M}$ of its metalogic modal program P_F. Let us now show the opposite direction, i. e., how a Kripke model $\mathcal{M}$ of a modal logic program P_F obtained by ontological encapsulation of the original many-valued logic program P_G, determines uniquely a many-valued model v of the logic program P_G. In fact, we define uniquely the mapping $v : H \rightarrow X$, as follows: for any modal atomic formula $\Diamond p_F(c_1, \ldots, c_n, \alpha)$, *true* in the Kripke model $\mathcal{M}$, we set $v(p(c_1, \ldots . c_n)) = \alpha$. It is easy to verify that such a definition of $v : H \rightarrow X$ is a Herbrand model of a many-valued logic program P_G. □

This transformation of multivalued logic programs into 2-valued multimodal logic programs can be briefly explained as follows: we transform the original multivalued atoms into the meta 2-valued atoms by enlarging the original atoms with a new logic attribute with the domain of truth values in X. This ontological encapsulation also eliminates the negation (in this case the negation $\sim$) by introducing a unary modal operator $\sim_F$. The remained binary multivalued lattice operations are substituted by the 2-valued binary modal operators with ternary accessibility relations.

Remark. In addition, this ontological encapsulation of logic programs into the *positive* (without the negation) modal programs, can be used, with some opportune modifications of the definitions above where a ground atom $p_F(c_1, \ldots, c_n, \alpha) \in H_F$ is true only for exactly one value $\alpha \in X$, to deal with the *inconsistency* of 2-valued logic programs: the resulting positive modal program is a *paraconsistent* logic program, i. e., for any given ground atom $p(c_1, \ldots, c_n)$ of the original 2-valued logic program that is inconsistent (both true and false), in the transformed *consistent* positive modal program we can (consistently) have two true ground atoms, $p_F(c_1, \ldots, c_n, 1)$ and $p_F(c_1, \ldots, c_n, 0)$. □

This knowledge invariant transformation of the many-valued logic into 2-valued (encapsulated) "meta" modal logic is provided as an example for interested readers to consider the bisimulation between these two modal logics.

7 Internal symmetry and data integration theory

7.1 DB (Database) category

Based on an observational point of view for relational databases, based on the power-view operator T introduced in Section A.3.2 in the Appendix, and on the theory of R-algebras for mapping operads, presented in Section A.3.5 by Definition 66 in the Appendix, we are able to introduce a category **DB** [61] for instance-databases and view-based mappings between them, with the set of its objects $\mathrm{Ob}_{\mathrm{DB}}$ and the set of its morphisms $\mathrm{Mor}_{\mathrm{DB}}$.

For the instance-database mappings, in Section A.3.4 and Example 31 in the Appendix, we considered the R-algebras that transform the set of schema-mapping operad's operations $\mathbf{M}_{AB} = \mathrm{MakeOperads}(\mathcal{M}_{AB}) = \{q_1, \ldots, q_k, 1_{r_\emptyset}\}$, of a given schema mapping $\mathcal{M}_{AB} : \mathcal{A} \to \mathcal{B}$, into the set of functions[1] $\alpha^*(\mathbf{M}_{AB}) = \{\alpha(q_1), \ldots, \alpha(q_k), q_\perp\}$. Consequently, each morphism in the base denotational **DB** category has to be a set of functions from an instance-database A into an instance-database B (of schemas $\mathcal{A}$ and $\mathcal{B}$, respectively).

However, we need to characterize such an instance-mapping morphism also by its information flux (introduced in Definition 67 in the Appendix) corresponding to this set of functions obtained from a mapping interpretation α (in Definition 66) of operad's operations q_i above.

In order to facilitate the presentation of the properties of this category, we will begin with simple objects and simple arrows, and then extend all to complex objects and complex arrows. In this section, we will establish the principal properties for the objects and morphisms in **DB**, and after that we will consider in dedicated sections the other particular properties.

Theorem 10. *Let us define, for a fixed enough big universe* $\mathcal{U} = \mathbf{dom} \bigcup \mathrm{SK}$ *where* **dom** *is a finite set of values and* $\mathrm{SK} = \{\omega_0, \omega_1, ., , \}$ *infinite set of indexed Skolem constants, the sets of objects* $\mathrm{Ob}_{\mathrm{DB}}$ *and morphisms* $\mathrm{Mor}_{\mathrm{DB}}$ *of this* **DB** *category:*

1. *A simple object is an instance-database A, composed of an set of n-ary (finite* $n \geq 0$*) relations* $R_i = a_i \in A$*,* $i = 1, 2, \ldots$ *also called "elements of A" with unique nullary empty relation* $\perp \in A$ *such that* $A \bigcap \perp^0 = \perp^0 = \{\perp\}$*.*
   Complex objects are obtained by disjunctive union of simple objects.
   A closed object *is an instance-database A such that* $A = TA \triangleq T(A)$*.*
   By Υ*, we denote the top-closed object in* **DB***, such that* $\Upsilon = \bigcup_{A \in \mathrm{Ob}_{\mathrm{DB}}} A$*.*

1 We use the algorithm MakeOperads in Section A.3.4 in order to transform logical schema mappings $\mathcal{M}_{AB} = \{\Phi\} : \mathcal{A} \to \mathcal{B}$ given by the SOtgds Φ (in Definition 61 of Section A.3.4 in the Appendix) into the algebraic operads $\mathbf{M}_{AB} = \mathrm{MakeOperad}(\mathcal{M}_{AB}) = \{v_1 \cdot q_{A,1}, \ldots, v_n \cdot q_{A,1}, 1_{r_\emptyset}\} : \mathcal{A} \to \mathcal{B}$.

https://doi.org/10.1515/9783111081670-007

2. *Let us define the set:* $S_{\mathrm{DB}} = \bigcup_{A,B\in \mathrm{Ob}_{\mathrm{DB}}}\{(\alpha^*(\mathbf{M}_{AB}), \mathrm{Flux}(\alpha, \mathbf{M}_{AB})) \mid \mathbf{M}_{AB} : \mathcal{A} \to \mathcal{B}$ *is a simple atomic sketch's mapping or a complex* $\biguplus$*-atomic sketch's mapping, and* α *is a mapping interpretation such that* $A = \alpha^*(\mathcal{A})$, $B = \alpha^*(\mathcal{B})\}$.
   The set of morphisms includes atomic morphisms in $\pi_1(S_{\mathrm{DB}})$ *and all other morphisms that can be obtained by well-defined compositions of atomic morphisms.*
3. *We define recursively the mapping* $B_\top : \mathrm{Mor}_{\mathrm{DB}} \to \mathrm{Ob}_{\mathrm{DB}}$ *such that for any simple morphism* $f : A \to B$*:*

$$B_\top(f) \triangleq \widetilde{f} \text{ if } (f, \widetilde{f}) \in S_{\mathrm{DB}};$$
$$B_\top(g) \bigcap B_\top(h) \text{ otherwise } (\text{when } f = g \circ h, h, g \in \mathrm{Mor}_{\mathrm{DB}} \text{ with } \mathrm{cod}(h) = \mathrm{dom}(g)).$$

These sets of objects and morphisms define the principal part of database category **DB**. *The extension of the mapping* $B_\top$ *to complex arrows (and their compositions) obtained by structural operators* $_ \biguplus _$, $[_ , _]$, $\langle _ , _ \rangle$ *and* $\lceil _ , _ \rfloor$ *is given in this section by Definition* 41.

Proof. For each simple object $A = \alpha^*(\mathcal{A})$, we have its identity morphism $\mathrm{id}_A : A \to A$ where $\mathrm{id}_A = \alpha^*(\mathrm{MakeOperads}(\{\bigwedge\{\forall \mathbf{x}_i (r_i(\mathbf{x}_i) \Rightarrow r_i(\mathbf{x}_i)) \mid r_i \in \mathcal{A}\}\})) = \alpha^*(\{1_{r_i} \in O(r_i, r_i) \mid r_i \in \mathcal{A}\} \bigcup \{1_{r_\emptyset}\}) = \{\mathrm{id}_{r_i} : \alpha(r_i) \to \alpha(r_i) \mid r_i \in \mathcal{A}\} \bigcup \{q_\perp\}$ and $\widetilde{\mathrm{id}_A} = TA$. For a complex object $A = \alpha^*(\mathcal{A}_1 \dagger \cdots \dagger \mathcal{A}_k) = \biguplus_{1\le i\le k} A_i$, where $A_i = \alpha^*(\mathcal{A}_i)$, its identity arrow is $\mathrm{id}_A = \biguplus_{1\le i\le k} \mathrm{id}_{A_i}$ with simple atomic identity arrows $\mathrm{id}_{A_i} : A_i \to A_i$. Moreover, $\widetilde{\mathrm{id}_A} = TA = \biguplus_{1\le i\le k} \widetilde{\mathrm{id}_{A_i}} = \biguplus_{1\le i\le k} TA_i$, which is a closed object as well.

Let $\mathbf{M}_{AB} = \mathrm{MakeOperads}(\mathcal{M}_{AB}) = \{q_1^A, \ldots, q_m^A, 1_{r_\emptyset}\}$ and

$$\mathbf{M}_{BC} = \mathrm{MakeOperads}(\mathcal{M}_{BC}) = \{q_1^B, \ldots, q_n^B, 1_{r_\emptyset}\}.$$

Then

$$\begin{aligned}
&\alpha^*(\mathbf{M}_{BC} \circ \mathbf{M}_{AB}) \\
&= \alpha^*\Big(\{q_i^B \cdot (q_{i_1}^A, \ldots, q_{i_{k_i}}^A) \mid 1 \le i \le n \text{ and } 1 \le i_j \le m \text{ for } 1 \le j \le k_i\} \bigcup \{1_{r_\emptyset} \cdot 1_{r_\emptyset}\}\Big) \\
&= (\text{from the definition of R-algebras}) \\
&= \{\alpha(q_i^B)(\alpha(q_{i_1}^A) \times \cdots \times \alpha(q_{i_{k_i}}^A)) \mid 1 \le i \le n \text{ and } 1 \le i_j \le m \text{ for } 1 \le j \le k_i\} \bigcup \{q_\perp\} \\
&= (\text{from } \alpha(1_{r_\emptyset}) = q_\perp) \\
&= \{\alpha(q_1^B), \ldots, \alpha(q_n^B), q_\perp\} \circ \{\alpha(q_1^A), \ldots, \alpha(q_M^A), q_\perp\} \\
&= \alpha^*(\{q_1^B, \ldots, q_n^B, 1_{r_\emptyset}\}) \circ \alpha^*(\{q_1^A, \ldots, q_m^A, 1_{r_\emptyset}\}) = \alpha^*(\mathbf{M}_{BC}) \circ \alpha^*(\mathbf{M}_{AB}).
\end{aligned}$$

Thus, for given two morphisms $f = \alpha^*(\mathbf{M}_{AB}) : A \to B$ and $g = \alpha^*(\mathbf{M}_{BC}) : B \to C$ their composition is equal t

$$g \circ f = \alpha^*(\mathbf{M}_{BC}) \circ \alpha^*(\mathbf{M}_{AB}) = \alpha^*(\mathbf{M}_{BC} \circ \mathbf{M}_{AB}) : A \to C.$$

Let as show the associativity of composition of morphisms in **DB**, with $h = \alpha^*(\mathbf{M}_{CD})$:

$$\begin{aligned} h \circ (g \circ f) &= \alpha^*(\mathbf{M}_{CD}) \circ \big(\alpha^*(\mathbf{M}_{BC}) \circ \alpha^*(\mathbf{M}_{AB})\big) = \alpha^*\big(\mathbf{M}_{CD} \circ (\mathbf{M}_{BC} \circ \mathbf{M}_{AB})\big) \\ &\quad \text{(from the associativity of the operad-mapping composition)} \\ &= \alpha^*\big((\mathbf{M}_{CD} \circ \mathbf{M}_{BC})\big) \circ \alpha^*(\mathbf{M}_{AB}) = \big(\alpha^*(\mathbf{M}_{CD}) \circ \alpha^*(\mathbf{M}_{BC})\big) \circ \alpha^*(\mathbf{M}_{AB}) \\ &= (h \circ g) \circ f. \end{aligned}$$

The left and right compositions of $\biguplus$-atomic arrows in Definition 69, in Section A.3.5 in the Appendix, are based on component-to-component composition of atomic arrows and are specified in detail in the rest of this section. Thus, the properties above hold for the $\biguplus$-atomic arrows as well. Consequently, **DB** is a well-defined category. □

The object composed of the empty relation only is denoted by $\perp^0$ and $T\perp^0 = \perp^0 = \{\perp\}$. It is shown that the empty database (a database with only empty relations) is isomorphic to this bottom object $\perp^0$. Notice that this theorem introduces, for each two simple objects A and B, the morphism $\perp^1 = \alpha^*(\mathbf{M}_{A,B})$ where $\mathbf{M}_{A,B} = \text{MakeOperads}(\{r_\emptyset \Rightarrow r_\emptyset\}) = \{1_{r_\emptyset}\}$, with the empty information flux $\widetilde{\perp^1} = \perp^0$ (from Definition 67 in Section A.3.5 in the Appendix). The objects $\perp^0$ and Υ are the bottom- and the top-closed objects in **DB**.

From definitions of morphisms, each morphism $f \in \text{Mor}_{\text{DB}}$ is a *set* of functions in $\alpha^*(\mathbf{M}_{AB})$, such that $\alpha(q_i) = \alpha(v_i)(\alpha(q_{A,i}))$ is a function that satisfies the mapping-interpretation conditions of Definition 66. Consequently, **DB** is different from the **Set** category (where each morphism is a single function). In the algorithm MakeOperads, each operad's operation q_i of a schema mapping between a schema $\mathcal{A}$ and a schema $\mathcal{B}$ is generally composed of two components: the first one corresponds to conjunctive query $q_{A,i}$ over a source database $\mathcal{A}$ that defines this view-based mapping and the second component v_i defines which contribution of this mapping is transferred into the target relation, i. e., a kind of Global-or-Local-As-View (GLAV) mapping (sound or exact) [92].

Based on the atomic schema mappings, we can introduce the *atomic morphisms* for instance-mappings in **DB** category as follows.

Definition 39 (Atomic morphisms in **DB** category). An atomic morphism in **DB** is a set of functions $f = \alpha^*(\mathbf{M}_{AB})$ with the information flux $\widetilde{f} = B_\top(f)$ such that $\mathbf{M}_{AB} = \{q_1, \ldots, q_k, 1_{r_\emptyset}\} : \mathcal{A} \to \mathcal{B}$ is an *atomic sketch's* schema mapping and α is a mapping interpretation (Definition 66 in Section A.3.5 in the Appendix) with $A = \alpha^*(\mathcal{A})$, $B = \alpha^*(\mathcal{B})$.

Let the schema mapping $\mathcal{M}_{AB} = \text{InverseOperads}(\mathbf{M}_{AB}) : \mathcal{A} \to \mathcal{B}$ be satisfied by α. Then for each operad's operation $q_i = v_i \cdot q_{A,i} \in \mathbf{M}_{AB}$, with $q_{A,i} \in O(r_{i1}, \ldots, r_{\text{im}}, r_q)$ and $v_i \in O(r_q, r_i)$, $r_i \in \mathcal{B}$, the function $\alpha(v_i)$ is an injection used to distinguish sound and exact assumption on the views as follows:

1. *Sound* case, i. e., when $\alpha(r_q) \subseteq \alpha(r_i)$.
2. *Exact* case, i. e., special inclusion case when $\alpha(r_q) = \alpha(r_i)$.

We extend the operators ∂_0 and ∂_1 to **DB** morphisms as well, so that $\partial_k(f) = \bigcup_{q_i \in \alpha^*(\mathbf{M}_{AB})} \partial_k(q_i)$, for $k = 0, 1$.

A *complete morphism* (c-morphism) $f : A \to B$ satisfies the condition $\partial_0(f) \subseteq A$.

Every atomic morphism is a complete morphism. Thus, each view-map q_{A_i}, i. e., an atomic morphism $f = \{q_{A_i}, q_\perp\} : A \longrightarrow TA$, is a complete morphism (the case when $B = TA$ and $\alpha(v_i)$ belongs to the "exact case").

As we have seen from Definition 69, in Section A.3.5 in the Appendix, for the $\biguplus$-atomic sketch's mappings, $\alpha^*(\biguplus(\mathbf{M}_{DA_1}, \ldots, \mathbf{M}_{DA_n})) = \biguplus(\alpha^*(\mathbf{M}_{DA_1}), \ldots, \alpha^*(\mathbf{M}_{DA_n}))$, and hence, they are compositions of atomic simple morphisms as well.

Remark. In the rest of this book, we will use $\mathcal{A} = (S_A, \Sigma_A)$ also for the set of relational symbols in S_A and an instance-database A for $\alpha^*(\mathcal{A}) = \alpha^*(S_A)$: it will simply be called a "database" when it is clear from the context. The functions ∂_0 and ∂_1 are different from dom and cod functions used for the category arrows. For an *atomic* morphism, or more generally, for a *complete* morphism $f : A \to B$, the $\partial_0(f)$ specifies exactly the subset of relations in a database A used by f, while $\partial_1(f)$ defines the target relations in a database B for this mapping. Thus, for a complete mapping $\partial_0(f) \subseteq \mathrm{dom}(f) = A$ and $\partial_1(f) \subseteq \mathrm{cod}(f) = B$. In the case when f is a simple view mapping, $\partial_1(f)$ is a singleton.

The fact that each atomic arrow $f : A \to B$ is a set of functions and has the information flux $\widetilde{f} = B_\top(f)$ is important for the composition of the arrows in **DB**: any two simple arrows $f, g : A \to B$ are equal ($f \equiv g$) not only when two sets of functions f and g are equal sets but also when they are different, but the information fluxes $\widetilde{f}$ and $\widetilde{g}$ are equal. More about this for complex arrows will be presented in Definition 44. □

From the proof of Theorem 10, for the composition of any two morphisms in **DB**,

$$
\begin{aligned}
&f = \alpha^*(\mathbf{M}_{AB}) : A \to B \quad \text{and} \quad g = \alpha^*(\mathbf{M}_{BC}) : B \to C,\\
&h = g \circ f = \alpha^*(\mathbf{M}_{BC}) \circ \alpha^*(\mathbf{M}_{AB}) = \alpha^*(\mathbf{M}_{BC} \circ \mathbf{M}_{AB})\\
&\quad = \{\alpha(q_i^B)(\alpha(q_{i_1}^A) \times \cdots \times \alpha(q_{i_{k_i}}^A)) \mid 1 \le i \le n, \text{ and } 1 \le i_j \le m \text{ for } 1 \le j \le k_i\} \bigcup \{\alpha(1_{r_\emptyset})\}\\
&\quad = \{q_{B_i} \cdot (q_{Ai_1}, \ldots, q_{Ai_{k_i}}) \mid 1 \le i \le n \text{ and } 1 \le i_j \le m \text{ for } 1 \le j \le k_i\} \bigcup \{q_\perp\},
\end{aligned}
$$

where q_{B_i} denotes the function $\alpha(q_i^B)$, for $1 \le i \le n$ and q_{A_j} denotes the function $\alpha(q_j^A)$, for $1 \le j \le m$ and $\cdot$ denotes the composition of functions. Graphically, such a composition of morphisms can be represented as a composition of trees (see the examples below where the "graphical-tree" morphisms are denoted by $f_T, g_T, \ldots$).

Generally, a composed morphism $h : A \to C$ is not a complete morphism, i. e., it can be graphically represented by a general tree such that not all its leaves are in A. Such an "*incomplete*" morphism is called a (partial) p-arrow. A p-arrow corresponds to a morphism obtained from a schema mapping $\mathcal{M}_{AB} = \{\Phi\} : \mathcal{A} \to \mathcal{B}$ where Φ is a SOtgd such that at least one left-hand side of its implications has the characteristic-functional symbols (for the relations that are not in the source schema $\mathcal{A}$).

By the graphical tree-composition of two trees in Figure 7.1, f_T (incomplete) and g_T (complete), we obtain the tree h_T (of the p-arrow $h = g \circ f : A \longrightarrow C$):

$$h_T = (g \circ f)_T = \bigcup_{q_{B_j} \in \alpha^*(\mathbf{M}_{BC})} \left\{ q_{B_j} \cdot \left(\bigcup_{q_{A_i} \in \alpha^*(\mathbf{M}_{AB}) \,\&\, \partial_1(q_{A_i}) \in \partial_0(q_{B_j})} \{q_{A_i}(\text{tree})\} \right) \right\}$$
$$= \{q_{B_j} \cdot \{q_{A_i}(\text{tree}) \mid \partial_1(q_{A_i}) \in \partial_0(q_{B_j})\} \mid q_{B_j} \in \alpha^*(\mathbf{M}_{BC})\}$$
$$= \{q_{B_j}(\text{tree}) \mid q_{B_j} \in \alpha^*(\mathbf{M}_{BC})\}$$

where $q_{A_i}(\text{tree})$ is the tree below the branch q_{A_i}.

The difference between this graphical tree-representation of morphisms in Figure 7.1 and of a standard operads-based representation is that in the graphical tree representation, for the relations in $\partial_0(h)$ that are not relations in $\partial_0(f)$ (called *hidden relations*), we do not use the identity functions $\alpha(1_r)$ for these hidden relations (relations that are not in the source database A but in some intermediate database). Let us consider the following example in order to facilitate the understanding of the composition of morphisms.

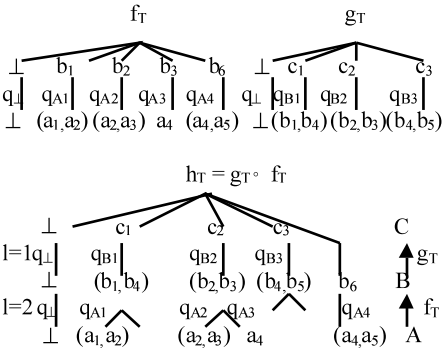


Figure 7.1: Composed tree.

Example 21. Let us consider the morphisms $f : A \longrightarrow B$ and $g : B \longrightarrow C$ such that $A = \{a_1, \ldots, a_6\}$, $B = \{b_1, \ldots, b_7\}$, $C = \{c_1, \ldots, c_4\}$, where $f = \{q_{A_1}, \ldots, q_{A_4}, q_\perp\}$, with $\partial_0(q_{A_1}) = \{a_1, a_2\}$, $\partial_0(q_{A_2}) = \{a_2, a_3\}$, $\partial_0(q_{A_3}) = \{a_4\}$, $\partial_0(q_{A_4}) = \{a_4, a_5\}$, $\partial_1(q_{A_1}) = \{b_1\}$, $\partial_1(q_{A_2}) = \{b_2\}$, $\partial_1(q_{A_3}) = \{b_3\}$, $\partial_1(q_{A_4}) = \{b_6\}$ and $g = \{q_{B_1}, \ldots, q_{B_3}, q_\perp\}$, with $\partial_0(q_{B_1}) = \{b_1, b_4\}$, $\partial_0(q_{B_2}) = \{b_2, b_3\}$, $\partial_0(q_{B_3}) = \{b_4, b_5\}$, $\partial_1(q_{B_1}) = \{c_1\}$, $\partial_1(q_{B_2}) = \{c_2\}$, $\partial_1(q_{B_3}) = \{c_3\}$. These two morphisms are represented by the trees f_T and g_T and their sequential composition by h_T in Figure 7.1.

From the point of view based on the information fluxes of these morphisms, the composition of morphisms $h = g \circ f : A \longrightarrow C$ can be graphically represented as *a part of the tree* h_T in Figure 7.2. It only provides the strict information contribution from the object A (i. e., the source) into the object C (target of this composed morphism).

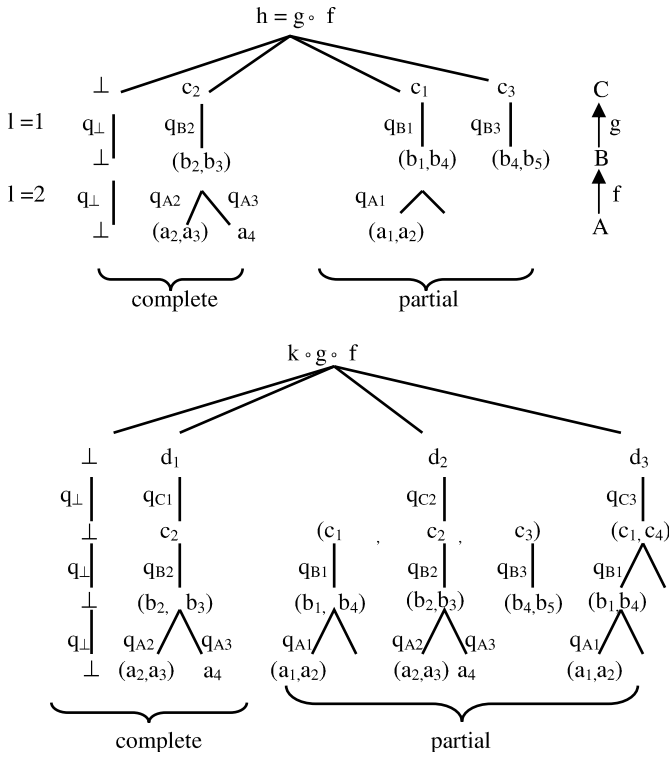


Figure 7.2: Obtained partial morphism.

Hence, $\partial_0(f) = \{a_1, a_2, a_3, a_4, a_5\}$, $\partial_1(f) = \{b_1, b_2, b_3, b_6\}$, $\partial_0(g) = \{b_1, b_2, b_3, b_4, b_5\}$, $\partial_1(g) = \{c_1, c_2, c_3\}$, while $\partial_0(h) = \partial_0(g \circ f) = \{a_1, a_2, a_3, a_4\} \neq \partial_0(f)$, $\partial_1(h) = \partial_1(g \circ f) = \{c_1, c_2, c_3\} = \partial_1(g)$.

Let us consider, e. g., the composition of the c-arrow $h : C \longrightarrow D$ with the composed arrow $g \circ f$ in the previous example, where $D = \{d_1, \ldots, d_4\}$, $h = \{q_{C_1}, q_{C_2}, q_{C_3}, q_\perp\}$, $\partial_0(q_{C_1}) = \{c_2\}$, $\partial_1(q_{C_1}) = \{d_1\}$, $\partial_0(q_{C_2}) = \{c_1, c_2, c_3\}$, $\partial_1(q_{C_2}) = \{d_2\}$, $\partial_0(q_{C_3}) = \{c_1, c_4\}$, $\partial_1(q_{C_3}) = \{d_3\}$, with $q_{B_2}(\text{tree}) = q_{B_2} \cdot \{q_{A_2}, q_{A_3}\}$ a complete and $q_{B_1}(\text{tree}) = q_{B_1} \cdot \{q_{A_1}, -\}$ a partial (incomplete) component of this tree, as represented in Figure 7.2. A composition of (complete) morphisms generally produces a partial (incomplete) morphism (only a part of the tree h_T represents a real contribution from A into C) with *hidden elements* (in the diagram of the composed morphism h, the elements b_4 and b_5 are a hidden relations).

In such a representation, for the consideration of the information fluxes, we "forgot" the parts of the tree $g_T \circ f_T$ that are not involved in a real information contribution of composed mappings from the source into the target object. In fact, the hidden relations do not appear on the left-hand side of SOtgd's implications of a given schema mapping, but only as the characteristic functions of these hidden relations. The hidden

relations appear always in the operad's operations and as a part of the domain of the set of functions that compose a given morphism $f : A \to B$. Consequently, operad's based representation of morphisms are equivalent to the logical representation based on the SOtgds. Based on Definition 67 in Section A.3.5 in the Appendix, the information flux for a composition of simple morphisms is given by the set-intersection of the information fluxes of all its simple morphisms.

Remark. The information flux $\widetilde{f} = B_{\top}(f)$ of a given morphism (instance-database mapping) $f : A \longrightarrow B$ is an instance-database as well; thus, an object in **DB**: the minimal information flux is equal to the bottom object $\perp^0$ so that, given any two database instances A and B in **DB**, there exists at least an arrow (morphism) between them $f : A \longrightarrow B$ such that $\widetilde{f} = \perp^0$ (e. g., $\perp^1 : A \to B$).

Definition 40. A complex object in **DB** is expressed by a disjoint union $\biguplus$ of simple instance-databases, and hence, each query over such a complex object can have the relational symbols belonging only to one of these simple databases. Let $A = A_1 \biguplus \cdots \biguplus A_m = \biguplus_{1\le j\le m} A_j = \{\{1\} \times A_1\} \bigcup \cdots \bigcup \{\{m\} \times A_m\}$ if $m \geq 2$; A_1 otherwise, and $B = \biguplus_{1\le i\le k} B_i = \{\{1\} \times B_1\} \bigcup \cdots \bigcup \{\{k\} \times B_k\}$ if $k \geq 2$; B_1 otherwise, where all A_j and B_i are simple (noncomposed) databases. Consequently, from the fact that all simple databases are separated, $T(\biguplus_{1\le j\le m} A_j) = \biguplus_{1\le j\le m} TA_j$.

(1) We say that A is *strictly complex* if $A_j \neq \perp^0$ for all $1 \le j \le m$.

(2) We can define the new composition $\otimes$ by $\Upsilon \otimes A = A \otimes \Upsilon = TA$, and

$$A \otimes B \triangleq \begin{cases} TA \bigcap TB, & \text{if } m = k = 1; \\ \biguplus_{1\le j\le m \ \&\ 1\le i\le k} (TA_j \bigcap TB_i), & \text{otherwise.} \end{cases}$$

(3) We define an "observational" PO relation $\preceq$ such that $\Upsilon \preceq B$ iff $B = \Upsilon$ and for $A = \biguplus_{1\le j\le m} A_j \neq \Upsilon$, $A \preceq B = \biguplus_{1\le i\le k} B_i$ iff there exists a mapping $\sigma : \{1, \ldots, m\} \to \{1, \ldots .k\}$ such that for each $1 \le j \le m$, $TA_j \subseteq TB_{\sigma(j)}$.

The equivalence in this ordering (i. e., $A \preceq B$ and $B \preceq A$) we denote by $A \approx B$.

Notice that $\preceq$ is a partial order (PO): $A \preceq A$ (for σ an identity function) and if $A \preceq B$ (with σ_1) and $B \preceq C$ (with σ_2) then $A \preceq C$ (with $\sigma = \sigma_2 \cdot \sigma_1$). For a given a tuple of objects (i. e., databases) in **DB** category $(S_1, \ldots, S_n), n \geq 2$, its disjoint union (from Definition 40) is $\biguplus_{1\le i\le n} S_i = \biguplus(S_1, \ldots, S_n) = \bigcup_{1\le i\le n} \{(i, a) \mid a \in S_i\}$ (the union *indexed by positions* of the sets in a given tuple), denoted also by $S_1 \biguplus S_2 \biguplus \cdots \biguplus S_n$. This indexing of simple databases that compose a complex object has to be maintained in the case when they are equal as well.

For example, a complex object $C \biguplus C$ has to be represented formally as an indexed complex object $A_1 \biguplus A_2$ with $A_1 = A_2 = C$. This "*indexing by position*" is fundamental for the formalization of complex morphisms between such complex objects. A disjoint union of two objects (databases) is isomorphic to a separation composition of these objects.

For any two simple morphisms $f = \alpha^*(\mathbf{M}_{AC}) = \{\alpha(q_1), \ldots, \alpha(q_k), q_\perp\} : A \to C$ and $g = \alpha^*(\mathbf{M}_{BD}) = \{\alpha(q'_1), \ldots, \alpha(q'_m), q_\perp\} : B \to D$, we define a complex morphism $(f \biguplus g) : A \biguplus B \to C \biguplus D$ by disjunctive union of them, i. e., by the set

$$\{(1, \alpha(q_i)) \mid q_i \in \mathbf{M}_{AC}\} \bigcup \{(2, \alpha(q'_j)) \mid q'_j \in \mathbf{M}_{BD}\} \bigcup \{q_\perp\} = \alpha^*(\mathbf{M}_{AC} \biguplus \mathbf{M}_{BD}).$$

Thus, from Definition 69, its flux is equal to

$$\begin{aligned} \widetilde{f \biguplus g} &= \alpha^*\left(\widetilde{\mathbf{M}_{AC} \biguplus \mathbf{M}_{BD}}\right) \\ &= \mathrm{Flux}(\alpha, \mathbf{M}_{AC}) \biguplus \mathrm{Flux}(\alpha, \mathbf{M}_{BD}) = \widetilde{f} \biguplus \widetilde{g} \in \mathrm{Ob}_{\mathrm{DB}}. \end{aligned}$$

Thus, we can consider the following extension of morphisms in **DB**, due to Theorem 10, based on Definition 69 in Section A.3.5 in the Appendix.

Definition 41. Complex arrows in **DB** presented by Theorem 10 are obtained by the structural operations $_ \biguplus _$, $[_, _]$, $\langle _, _ \rangle$ and $\lceil _, _ \rfloor$ (all A, B, C and D are simple objects), as follows:

1. Morphism $f \biguplus g : A \biguplus B \to C \biguplus D$ for any $f : A \to C$ and $g : B \to D$;
2. Morphism $[f, g] : A \biguplus B \to C$ for any $f : A \to C$ and $g : B \to C$;
3. Morphism $\langle f, g \rangle : A \to C \biguplus D$ for any $f : A \to C$ and $g : A \to D$;
4. Morphism $\lceil f \circ k, g \circ l \rfloor = [f, g] \circ \langle k, l \rangle : A \to B \biguplus D \to C$ with simple arrows $f \circ k : A \to B \to C$ and $g \circ l : A \to D \to C$, such that $\widetilde{f \circ k} \neq \perp^0$ and $\widetilde{g \circ l} \neq \perp^0$ and with reductions: $\lceil k, g \rfloor = k$ if $\widetilde{k} \neq \perp^0$ and $\widetilde{g} = \perp^0$; $\lceil k, g \rfloor = \perp^1$ if $\widetilde{k} = \widetilde{g} = \perp^0$.

Hence, a *simple* arrow (or morphism) is an arrow between two simple (nonseparation-composed) objects. Notice that a complex arrow $\lceil k, l \rfloor : A \to C$ exists only as a composition of two complex arrows defined in point 4. It means that both simple arrows $k, l : A \to C$ are composed arrows with *different* intermediate databases.

We recall that an arrow $\perp^1 = \{q_\perp : \perp \to \perp\} : A \to C$, which represents an empty function, means that, in fact, we have no mapping between A and B (the information flux of $\perp^1$ is empty, i. e., $\widetilde{\perp^1} = \perp^0 = \{\perp\}$). This fact explains why we are using only the arrows with nonempty fluxes in point 4 in a representation of a set of arrows between two fixed objects by $\lceil , \rfloor$. Based on Definitions 69 in Section A.3.5 in the Appendix, and 41 for binary compositions, we have the following cases for composition of complex morphisms (all A_i, B_i, C_i, $i = 1, 2$, and A, B, C are simple objects):

1. for $k : A_1 \to B_1$, $l : A_2 \to B_2$, $f : B_1 \to C_1$ and $g : B_2 \to C_2$,

$$\left(f \biguplus g\right) \circ \left(k \biguplus l\right) = (f \circ k) \biguplus (g \circ l) : A_1 \biguplus A_2 \to C_1 \biguplus C_2,$$

2. for $k : A \to B_1$, $l : A \to B_2$, $f : B_1 \to C_1$ and $g : B_2 \to C_2$,

$$\left(f \biguplus g\right) \circ \langle k, l \rangle = \langle f \circ k, g \circ l \rangle : A \to C_1 \biguplus C_2,$$

3. for $k : A_1 \to B_1, l : A_2 \to B_2, f : B_1 \to C$ and $g : B_2 \to C$,

$$[f,g] \circ \left(k \biguplus l\right) = [f \circ k, g \circ l] : A_1 \biguplus A_2 \to C,$$

4. for $k : A \to B_1, l : A \to B_2, f : B_1 \to C$ and $g : B_2 \to C$,

$$[f,g] \circ \langle k,l \rangle = \lceil [f \circ k, g \circ l] \rfloor, \quad (\text{if } f \circ k \neq \perp^1, g \circ l \neq \perp^1);$$
$$g \circ l \quad (\text{if } f \circ k = \perp^1, g \circ l \neq \perp^1);$$
$$f \circ k \quad (\text{if } f \circ k \neq \perp^1, g \circ l = \perp^1);$$
$$\perp^1 \text{ otherwise (an arrow from } A \text{ to } C),$$

5. for $k : A_1 \to B, f : B \to C_1$ and $g : B \to C_2$

$$\langle f,g \rangle \circ k = \langle f \circ k, g \circ k \rangle : A_1 \to C_1 \biguplus C_2,$$

6. for $k : A_1 \to B, l : A_2 \to B$ and $f : B \to C$,

$$f \circ [k,l] = [f \circ k, g \circ k] : A_1 \biguplus A_2 \to C,$$

7. for $k : A \to B, l : A \to B, f : B \to C_1$, and $g : B \to C_2$

$$\langle f,g \rangle \circ \lceil k,l \rfloor = \langle \lceil f \circ k, f \circ l \rfloor, \lceil g \circ k, g \circ l \rfloor \rangle : A \to C_1 \biguplus C_1,$$

8. for $k : A_1 \to B, l : A_2 \to B, f : B \to C$ and $g : B \to C$,

$$\lceil f,g \rfloor \circ [k,l] = \left[\lceil f \circ k, g \circ k \rfloor, \lceil f \circ l, g \circ l \rfloor \right] : A_1 \biguplus A_2 \to C,$$

9. for $k : A \to B, l : A \to B, f : B \to C$ and $g : B \to C$,

$$\lceil f,g \rfloor \circ \lceil k,l \rfloor = \lceil f \circ k, g \circ k, f \circ l, g \circ l \rfloor : A \to C.$$

Now we can extend these binary structural operators in Definition 41 into n-ary operators.

Definition 42. We generalize the binary structure operators in Definition 41 into n-ary operators, for any $n \geq 2$, by:

1. $\biguplus_{1 \leq i \leq n} f_i = \biguplus(f_1, \ldots, f_n) : \biguplus_{1 \leq i \leq n} A_i \to \biguplus_{1 \leq i \leq n} B_i$ where $f_i : A_i \to B_i$ for each $1 \leq i \leq n$.
2. $[f_1, \ldots, f_n] : \biguplus_{1 \leq i \leq n} A_i \to B$, where $f_i : A_i \to B$ for each $1 \leq i \leq n$.
3. $\langle f_1, \ldots, f_n \rangle : A \to \biguplus_{1 \leq i \leq n} B_i$, where $f_i : A \to B_i$ for each $1 \leq i \leq n$.
4. $\lceil f_1, \ldots, f_n \rfloor : A \to B$, where $\widetilde{f_i} \neq \perp^0$ for each $1 \leq i \leq n$.

Remark. In order to simplify the presentation, we extend the structure operators also to unary cases: for example, $\biguplus_{1 \leq i \leq 1} A_i = \biguplus(A_1) = A_1$, $\biguplus_{1 \leq i \leq 1} f_i = \biguplus(f_1) = f_1$.

We extend the application of $\biguplus$ also to (unordered) sets, so that for a given nonempty set (of sets) S, $\biguplus S$ represents an application of $\biguplus$ to *any* ordered list (tuple) of sets in S. The operator $\biguplus$ is a commutative and associative binary operator (up to isomorphism) so that, given any ordered list (a permutation) $(s_1, \ldots, s_n)$ of all elements (which are the sets) in S, the following isomorphism is valid in **DB**:

$$\biguplus S \simeq \biguplus(s_1, \ldots, s_n) = \left(s_1 \biguplus \cdots \biguplus s_n\right) = \bigcup_{1 \le i \le n} \{(i, a) \mid a \in s_i\}.$$

Thus, e. g., we have the following cases for a composition of complex morphisms obtained by using n-ary separation compositions:

1. For $\biguplus_{1 \le i \le n} f_i : \biguplus_{1 \le i \le n} A_i \to \biguplus_{1 \le i \le n} B_i$ and $\biguplus_{1 \le i \le n} g_i : \biguplus_{1 \le i \le n} B_i \to \biguplus_{1 \le i \le n} C_i$,

$$\left(\biguplus_{1 \le i \le n} g_i\right) \circ \left(\biguplus_{1 \le i \le n} f_i\right) = \biguplus_{1 \le i \le n} (g_i \circ f_i) : \biguplus_{1 \le i \le n} A_i \to \biguplus_{1 \le i \le n} C_i$$

2. For $\langle f_1, \ldots, f_n \rangle : A \to \biguplus_{1 \le i \le n} B_i$ and $\biguplus_{1 \le i \le n} g_i : \biguplus_{1 \le i \le n} B_i \to \biguplus_{1 \le i \le n} C_i$,

$$\left(\biguplus_{1 \le i \le n} g_i\right) \circ \langle f_1, \ldots, f_n \rangle = \langle g_1 \circ f_1, \ldots, g_n \circ f_n \rangle : A \to \biguplus_{1 \le i \le n} C_i$$

3. For $\biguplus_{1 \le i \le n} f_i : \biguplus_{1 \le i \le n} A_i \to \biguplus_{1 \le i \le n} B_i$ and $[g_1, \ldots, g_n] : \biguplus_{1 \le i \le n} B_i \to C$,

$$[g_1, \ldots, g_n] \circ \left(\biguplus_{1 \le i \le n} f_i\right) = [g_1 \circ f_1, \ldots, g_n \circ f_n] : \biguplus_{1 \le i \le n} A_i \to C$$

4. For $\langle f_1, \ldots, f_n \rangle : A \to \biguplus_{1 \le i \le n} B_i$ and $[g_1, \ldots, g_n] : \biguplus_{1 \le i \le n} B_i \to C$,

$$[g_1, \ldots, g_n] \circ \langle f_1, \ldots, f_n \rangle = \lceil S \rfloor : A \to C, \quad \text{if } |S| \ge 1; \quad \perp^1 : A \to C \text{ otherwise}$$

(where $S = \{g_i \circ f_i \mid \widetilde{g_i \circ f_i} = \widetilde{g_i} \bigcap \widetilde{f_i} \neq \perp^0, 1 \le i \le n\}$)

5. For $f : A \to B$ and $\langle g_1, \ldots, g_n \rangle : B \to \biguplus_{1 \le i \le n} C_i$,

$$\langle g_1, \ldots, g_n \rangle \circ f = \langle g_1 \circ f, \ldots, g_n \circ f \rangle : A \to \biguplus_{1 \le i \le n} C_i$$

6. For $[f_1, \ldots, f_n] : \biguplus_{1 \le i \le n} A_i \to B$ and $g : B \to C$,

$$g \circ [f_1, \ldots, f_n] = [g \circ f_1, \ldots, g \circ f_n] : \biguplus_{1 \le i \le n} A_i \to C$$

7. For $f_i : A \to B, 1 \le i \le n$ and $g : B \to C$,

$$g \circ \lceil f_1, \ldots, f_n \rfloor = \lceil S \rfloor : A \to C, \quad \text{if } |S| \ge 1; \quad \perp^1 : A \to C \text{ otherwise}$$

(where, from Definition 41, $S = \{g \circ f_i \mid \widetilde{g \circ f_i} \neq \perp^0, 1 \le i \le n\}$)

8. For $g_i : B \to C, 1 \le i \le n$ and $f : A \to B$,

$$\lceil g_1, \ldots, g_n \rfloor \circ f = \lceil S \rfloor : A \to C, \quad \text{if } |S| \ge 1; \quad \perp^1 : A \to C \text{ otherwise}$$

(where, from Definition 41, $S = \{g_i \circ f \mid \widetilde{g_i \circ f} \neq \perp^0, 1 \le i \le n\}$).

By these rules of composition of complex arrows, we obtain that the resulting arrow is a composition of structural-operators $\biguplus(_, \ldots, _)$, $[_, \ldots, _]$, $\langle _, \ldots, _ \rangle$ and $\lceil _, \ldots, _ \rfloor$.

The "point-to-point" (ptp) arrow is *nonempty* (with flux different from $\perp^0$) simple arrow (a path of only simple atomic arrows), between two simple objects that compose the source and target database of this complex arrow: if we have more than one simple arrow between the same source and target simple objects, then we fuse them (by union) into a single ptp arrow, as follows.

Definition 43. For any complex arrow $h : A \to B$ between "indexed by position" complex objects $A = \biguplus_{1 \le j \le m} A_j$ and $B = \biguplus_{1 \le i \le k} B_i, m, k \ge 1$, we define its set of ptp morphisms by

$$\frac{h}{\biguplus} \triangleq \Big\{ h_{ji} = \bigcup \{ f_l : A_j \to B_i \mid f_l \text{ is a composition of simple arrows in } h \text{ such that } \widetilde{f_l} \neq \perp^0 \} \neq \emptyset \mid 1 \le j \le m, 1 \le i \le k \Big\}.$$

Then we extend the mapping $B_\top$ to complex arrows also: $B_\top(h) = \widetilde{h}$ is equal (up to isomorphism) to the object

$$\widetilde{\biguplus \frac{h}{\biguplus}} \triangleq \begin{cases} \biguplus\{B_\top(h_{ji}) \mid h_{ji} \in \frac{h}{\biguplus}\} = \biguplus\{\widetilde{h_{ji}} \mid h_{ji} \in \frac{h}{\biguplus}\}, & \text{if } |\frac{h}{\biguplus}| \ge 2; \\ B_\top(h_{ji}) = \widetilde{h_{ji}}, & \text{if } \frac{h}{\biguplus} = \{h_{ji}\}; \\ \perp^0, & \text{otherwise.} \end{cases}$$

Notice that for a given complex arrow h each of its ptp arrows has a nonempty information flux (different from $\perp^0$) so that a ptp arrow is different from $\perp^1$ arrow. In the previous example for the composition of complex arrows, the resulting arrow is composed of point-to-point arrows. For example, in rule 5, we obtained the complex arrow $\langle f \circ k, g \circ k \rangle : A_1 \to C_1 \biguplus C_2$ where $f \circ k : A_1 \to C_1$ and $g \circ k : A_1 \to C_2$ are two point-to-point simple arrows. In rule 3, we obtained the complex arrow $[f \circ k, g \circ l] : A_1 \biguplus A_2 \to C$ where $f \circ k : A_1 \to C$ and $g \circ l : A_2 \to C$ are two point-to-point simple arrows, etc.

This property extends by induction to any composition of complex arrows. The fact that the source and target complex objects A and B of a complex morphism $h : A \to B$ are "indexed by position" is fundamental for definition (indexing) of its ptp arrows. For example, the complex arrow $h = [\mathrm{id}_C, \mathrm{id}_C, \mathrm{id}_C] : C \biguplus C \biguplus C \to C$ has to be represented formally by $h : A_1 \biguplus A_2 \biguplus A_3 \to B_1$, with $A_1 = A_2 = A_3 = B_1 = C$, so that $\frac{h}{\biguplus} = \{h_{11} = \mathrm{id}_C : A_1 \to B_1, h_{21} = \mathrm{id}_C : A_2 \to B_1, h_{31} = \mathrm{id}_C : A_3 \to B_1\}$ is the set of three ptp arrows and not the singleton $\{\mathrm{id}_C : C \to C\}$. This requirement is fundamental

for the definition of ptp arrows because it takes in consideration the mutually separated databases in the complex source object A and mutually independent morphisms from them into the target database B. Based on Definitions 69 in Section A.3.5 in the Appendix, 41 and 42, for $\frac{h}{\uplus} \neq \emptyset$ we obtain $\widetilde{h} \simeq \biguplus\{\widetilde{f_i} \mid f_i \in \frac{h}{\uplus}\}$ (i. e., $\widetilde{h}$ is not equal to $\widetilde{\biguplus \frac{h}{\uplus}}$ but only *isomorphic* to it). This isomorphism (instead of an equality) is based on the fact that $\biguplus(A, B) = A \biguplus B \simeq B \biguplus A = \biguplus(B, A)$ (formally demonstrated by point 2 of the following Lemma 3) and ptp arrows in h, as explained in Definition 42, are ordered while set $\frac{h}{\uplus}$ is unordered: when $\biguplus$ is applied to a set, then *any* ordering can be taken in this set.

Notice that the mapping $B_{\top}$ for complex morphisms continues to generate the closed objects. In fact, we can show it recursively. Let $\widetilde{f} = B_{\top}(f)$, $\widetilde{g} = B_{\top}(g)$ be closed objects (i. e., from the fact that power-view operator T (Section A.3.2) is idempotent, $T\widetilde{f} = \widetilde{f}$, $T\widetilde{g} = \widetilde{g}$). Then $T(\widetilde{f} \biguplus \widetilde{g}) = T\widetilde{f} \biguplus T\widetilde{g} = \widetilde{f} \biguplus \widetilde{g}$, and hence, closed as well.

Remark. As we have specified in Definition 43, the path arrows with the same simple source and target objects are fused (by union) into a single morphism. For example, for given two simple databases A and B, the complex arrows $\lceil f, g \rfloor : A \to B$, where $f = \alpha(\mathbf{M}_{AB}^{(1)})$ and $g = \alpha(\mathbf{M}_{AB}^{(2)})$ (with sketch's arrows $\mathbf{M}_{AB}^{(n)} = \text{MakeOperads}(\mathcal{M}_{AB}^{(n)}) : \mathcal{A} \to \mathcal{B}$, $n = 1, 2$ and $\mathcal{M}_{AB}^{(1)} = \{\Phi\} : \mathcal{A} \to \mathcal{B}$, $\mathcal{M}_{AB}^{(2)} = \{\Psi\} : \mathcal{A} \to \mathcal{B}$), will be fused into a single arrow $k : A \to B$, where $k = f \bigcup g = \alpha(\mathbf{M}_{AB})$ is obtained by fusing the sketch's mappings $\mathbf{M}_{AB}^{(1)}$ and $\mathbf{M}_{AB}^{(1)}$ into $\mathbf{M}_{AB} = (\mathbf{M}_{AB}^{(1)} \bigcup \mathbf{M}_{AB}^{(1)}) : \mathcal{A} \to \mathcal{B}$. Thus, based on Definition 67 in Section A.3.5 in the Appendix, the flux of $f \bigcup g$ is equal to (as in point 3 of Definition 69):

$$\begin{aligned}
\widetilde{\lceil f, g \rfloor} = \widetilde{f \bigcup g} &= T\Big(\Delta(\alpha, \mathcal{M}_{AB}^{(1)}) \bigcup \Delta(\alpha, \mathcal{M}_{AB}^{(2)})\Big) \\
&= T\Big(T(\Delta(\alpha, \mathcal{M}_{AB}^{(1)})) \bigcup T(\Delta(\alpha, \mathcal{M}_{AB}^{(2)}))\Big) = T\Big(\text{Flux}(\alpha, \mathcal{M}_{AB}^{(1)}) \bigcup \text{Flux}(\alpha, \mathcal{M}_{AB}^{(2)})\Big) \\
&= T\Big(\widetilde{f} \bigcup \widetilde{g}\Big) \neq \widetilde{f} \biguplus \widetilde{g}.
\end{aligned}$$

□

Generally, when $g = f$, the complex arrow $\lceil f, g \rfloor : A \to B$ may represent the two composed independent mapping paths f and g with different intermediate databases. Thus, $\widetilde{\lceil f, g \rfloor} = T(\widetilde{f} \bigcup \widetilde{f}) = T\widetilde{f} = \widetilde{f}$ (a closed object), i. e., this pair of arrows $\lceil f, g \rfloor$ transfers the same information flux from the source to target database as a single mapping $f : A \to B$. However, they cannot be considered equal: the *redundant* transferring of the same information flux represents a more robust database mapping system in which if the first mapping is interrupted by a failure in one of its intermediate databases then the second (alternative) mapping may continue to work and to guarantee the general security mapping requirements.

This database mapping system requirement explains that, in order to establish the equality of two mappings $k = \lceil f, f \rfloor : A \to B$ and $f : A \to B$, their observational equality (when they have equal information fluxes, which are the set of all transferred views from A into B) is not a sufficient condition.

In fact, let us consider the mappings $\langle l_1, l_2 \rangle : A \to C \biguplus C$ and $[m_1, m_2] : C \biguplus C \to B$, so that their composition is $\lceil f_1, f_1 \rfloor = [m_1, m_2] \circ \langle l_1, l_2 \rangle = \lceil m_1 \circ l_1, m_2 \circ l_2 \rfloor : A \to B$, also

when $m_1 = m_2 = m$ and $l_1 = l_2 = l$, is different from a single mapping $f = m \circ l : A \to B$, i. e., $\lceil f, f \rfloor \neq f$. However, if $m_2 = \perp^1$ or $l_2 = \perp^1$, then $\lceil m_1, m_2 \rfloor \circ \langle l_1, l_2 \rangle = \lceil f_1, \perp^1 \rfloor = f_1$ because the alternative mapping $\perp^1 : A \to B$ has an empty information flux (it does not transfer anything from A into B) and can be eliminated from the mappings (as specified by point 2.4 in Definition 41).

Lemma 1. *For any complex arrow $h : A \to B$ with $A = \biguplus_{1 \leq j \leq m} A_j$, $m \geq 1$ and $B = \biguplus_{1 \leq i \leq k} B_i$, $k \geq 1$, any its substructure $\lceil h_{l_1}, \ldots, h_{l_n} \rfloor : A_j \to B_i$, $n \geq 2$, where each h_{l_i} is a simple arrow (with simple source and target objects) is fused into a single ptp arrow $(h_{ji} : A_j \to B_i) \in \frac{h}{\biguplus}$. Thus, its information flux is $\widetilde{h_{ji}} = T(\bigcup \{\widetilde{h_{l_i}} \mid 1 \leq i \leq n\})$. Given any pair of simple source and target databases A_j and B_i, there is at most one ptp arrow $h_{ji} \in \frac{h}{\biguplus}$ between them.*

Proof. If each h_{l_i} is a simple arrow with simple source and target objects (databases) A_j and B_i, respectively, then $\lceil h_{l_1}, \ldots, h_{l_n} \rfloor : A_j \to B_i$ is fused into a single ptp, as it directly follows from definition of $\frac{h}{\biguplus}$. The flux of these fused (by union) arrows into one ptp arrow has then the information flux as explained above and is specified in the previous remark.

The fact that, for any pair of simple source and target databases A_j and B_i, there is at most one ptp arrow $h_{ji} \in \frac{h}{\biguplus}$ between them is a direct consequence of the definition of $\frac{h}{\biguplus}$. It means that h cannot have more than $k \cdot m$ ptp arrows. □

The logical equivalence of the SOtgds of two schema mappings is an insufficient requirement for the equality of the mappings at instance-level, and we need a more appropriate refinement of it based on the information fluxes. For any two logically equivalent schema mappings $\mathcal{M}_{AB} = \{\Phi\} : \mathcal{A} \to \mathcal{B}$ and $\mathcal{M}'_{AB} = \{\Phi'\} : \mathcal{A} \to \mathcal{B}$, where the SOtgds Φ and Φ' are logically equivalent, for any mapping-interpretation α we will have that the morphism (a set of functions) $f = \alpha^*(\text{MakeOperads}(\mathcal{M}_{AB}))$ is *equal* to the set of functions $g = \alpha^*(\text{MakeOperads}(\mathcal{M}'_A B))$. It holds from the fact that operad's based representation of the schema mappings is equivalent to the logical representation given by SOtgds. However, this kind of strong equality is not useful in our case, and we are considering a more weak equality based on observational equivalence and information fluxes of instance-database mappings $f, g : A \to B$, where $A = \alpha^*(\mathcal{A})$ and $B = \alpha^*(\mathcal{B})$, as follows.

Definition 44 (Equality of arrows in **DB**). In the observational-based semantics, the equality '≡' of two morphisms $f, g : A \to B$ (used for the commutativity of diagrams) in **DB** is defined as follows:

1. If both arrows are simple, then $f \equiv g$ iff $\widetilde{f} = \widetilde{g}$;
2. If $f = \lceil f_1, \ldots, f_m \rfloor$ and $g = \lceil g_1, \ldots, g_k \rfloor$, with $m, k \geq 1$ and A and B simple databases, then $f \equiv g$ iff $m = k$ and there is a bijection $\sigma : \lceil f_1, \ldots, f_m \rfloor \to \lceil g_1, \ldots, g_k \rfloor$ such that $\widetilde{f_j} = \widetilde{\sigma(f_j)}$ for all $j = 1, \ldots, m$;
3. Otherwise, $f \equiv g$ iff $\frac{f}{\biguplus} = \frac{g}{\biguplus}$ (have the same set of ptp arrows).

Thus, if the morphisms f and g are equal in **DB**, they are not necessarily equal set-theoretically. By having in mind this difference between the common set-equality '=' and the **DB** arrows equality '≡', we will continue to use the same notation $f = g$ also for the arrows (instead of a more appropriate notation $f \equiv g$).

Consequently, the equality of arrows (morphisms) in **DB** is not the set-theoretical equality (by considering that each morphism is mathematically just a set of functions).

This observational equality of instance-database mappings means that at this level of DB abstraction, two instance-database mappings are equal iff the information transferred from the source instance-database into a target instance-database for both mappings is equivalent (or equal in special cases, because if $\widetilde{f} = \widetilde{g}$ then the identity morphism $\mathrm{id} : \widetilde{f} \to \widetilde{g}$ is an isomorphism as well, so that $\widetilde{f} \simeq \widetilde{g}$) and in cases 2 above satisfy some additional requirements.

Remark. In what follows, we will use the common symbol $'='$ for the *equality of morphisms* in **DB** category by considering this observational meaning (and not the set-theoretic meaning). For all other cases, the equality symbol $'='$ has the set-theoretical meaning (e. g., for the information fluxes $\widetilde{f} = B_{\top}(f) = \widetilde{g} = B_{\top}(g)$ or in definitions like of morphisms $f = \alpha^*(\mathrm{MakeOperads}(\mathcal{M}))$, the meaning for $'='$ is standard set-theoretic). All *commutativity diagrams* in **DB** will be based on this abstracted (nonset-based) equality for the morphisms.

Example 22. It is easy to verify the validity of composition rules for complex arrows (we denote by $=^n$ the equation obtained by application of the n-th rule of compositions of arrows defined previously):

1. Let us show for the case $k : A_1 \to B, l : A_2 \to B, f : B \to C_1$ and $g : B \to C_2$, that $h = \langle f, g\rangle \circ [k, l] = \langle [f \circ k, f \circ l], [g \circ k, g \circ l]\rangle = [\langle f \circ k, g \circ k\rangle, \langle f \circ l, g \circ l\rangle] : A_1 \biguplus A_2 \to C_1 \biguplus C_2$, with the four obtained point-to-point arrows $f \circ k : A_1 \to C_1$, $g \circ k : A_1 \to C_2$, $f \circ l : A_2 \to C_1$ and $g \circ l : A_2 \to C_2$. That is, $\frac{h}{\biguplus} = \{f \circ k, g \circ k, f \circ l, g \circ l\}$.

We have $\langle f, g\rangle \circ [k, l] =^5 \langle f \circ [k, l], g \circ [k, l]\rangle =^6 \langle [f \circ k, f \circ l], [g \circ k, g \circ l]\rangle$ and $\langle f, g\rangle \circ [k, l] =^6 [\langle f, g\rangle \circ k, \langle f, g\rangle \circ l] =^5 [\langle f \circ k, g \circ k\rangle, \langle f \circ l, g \circ l\rangle]$.

We can show that it is true by considering the ptp arrows. In fact,

$$\frac{h}{\biguplus} = \frac{\langle [f \circ k, f \circ l], [g \circ k, g \circ l]\rangle}{\biguplus} = \frac{[\langle f \circ k, g \circ k\rangle, \langle f \circ l, g \circ l\rangle]}{\biguplus}$$
$$= \{f \circ k : A_1 \to C_1, g \circ k : A_1 \to C_2, f \circ l : A_2 \to C_1, g \circ l : A_2 \to C_2\}.$$

2. Let us show that $f = \langle \mathrm{id}_C, \mathrm{id}_C\rangle \circ [\mathrm{id}_C, \mathrm{id}_C] = \langle [\mathrm{id}_C, \mathrm{id}_C], [\mathrm{id}_C, \mathrm{id}_C]\rangle : C \biguplus C \to C \to C \biguplus C$ is different from $g = \mathrm{id}_C \biguplus \mathrm{id}_C : C \biguplus C \to C \biguplus C$. In fact, in order to define the ptp arrows of these two morphisms, we have to use the "indexing by position" for the source database $A = A_1 \biguplus A_2$ and target database $B = B_1 \biguplus B_1$ where $A_1 = A_2 = B_1 = B_2 = C$. Thus, $\frac{\langle \mathrm{id}_C, \mathrm{id}_C\rangle \circ [\mathrm{id}_C, \mathrm{id}_C]}{\biguplus} = \{f_{11} = \mathrm{id}_C : A_1 \to B_1, f_{12} = \mathrm{id}_C : A_1 \to B_2, f_{21} =$

$$\mathrm{id}_C : A_2 \to B_1, f_{22} = \mathrm{id}_C : A_2 \to B_2\} \neq \{g_{11} = \mathrm{id}_C : A_1 \to B_1, g_{22} = \mathrm{id}_C : A_2 \to B_2\} = \frac{\mathrm{id}_C \biguplus \mathrm{id}_C}{\biguplus}.$$

Consequently, by point 3 of Definition 44, the identity arrow $\mathrm{id}_C \biguplus \mathrm{id}_C \neq \langle \mathrm{id}_C, \mathrm{id}_C \rangle \circ [\mathrm{id}_C, \mathrm{id}_C]$.

In fact, $\mathrm{id}_A = \mathrm{id}_{A_1} \biguplus \mathrm{id}_{A_2}$ because for any $f : A \to X$ we have from Definition 43 that $\frac{f \circ (\mathrm{id}_{A_1} \biguplus \mathrm{id}_{A_2})}{\biguplus} = \{f_{i_2} \circ \mathrm{id} \mid f_{i_2} \in \frac{f}{\biguplus}, \mathrm{id} \in \{\mathrm{id}_{A_1}, \mathrm{id}_{A_2}\} = \frac{\mathrm{id}_{A_1} \biguplus \mathrm{id}_{A_2}}{\biguplus}, \mathrm{dom}(f_{i_2}) = \mathrm{cod}(\mathrm{id})\} = \{f_{i_2} \mid f_{i_2} \in \frac{f}{\biguplus}\} = \frac{f}{\biguplus}$ so that, by point 3 of Definition 44, $f = f \circ (\mathrm{id}_{A_1} \biguplus \mathrm{id}_{A_2})$, and analogously, $f = (\mathrm{id}_{A_1} \biguplus \mathrm{id}_{A_2}) \circ f$, so that $(\mathrm{id}_{A_1} \biguplus \mathrm{id}_{A_2})$ is the identity arrow of $A = A_1 \biguplus A_2$. That is, $\mathrm{id}_A = \mathrm{id}_{A_1 \biguplus A_2} = \mathrm{id}_{A_1} \biguplus \mathrm{id}_{A_2}$.

Let us consider the associativity of composition of complex morphisms (that **DB** is well-defined for complex arrows as well):

3. For $k : A_1 \to B, l : A_2 \to B, f : B \to C_1, g : B \to C_2, j : C_1 \to D$ and $m : C_2 \to D$, we consider the composition $[j, m] \circ \langle f, g \rangle \circ [k, l] : A_1 \biguplus A_2 \to B \to C_1 \biguplus C_2 \to D$. Let us consider the two different orders of composition:

   3.1
   $$h_1 = [j, m] \circ (\langle f, g \rangle \circ [k, l]) =^5 [j, m] \circ \langle [f \circ k, f \circ l], [g \circ k, g \circ l] \rangle$$
   $$=^4 \lceil [j \circ f \circ k, j \circ f \circ l], [m \circ g \circ k, m \circ g \circ l] \rfloor : A_1 \bigbiguplus A_2 \to D,$$

   3.2
   $$h_2 = ([j, m] \circ \langle f, g \rangle) \circ [k, l] =^4 \lceil [j \circ f, m \circ g] \circ [k, l]$$
   $$=^7 \lceil [j \circ f \circ k, j \circ f \circ l], [m \circ g \circ k, m \circ g \circ l] \rfloor : A_1 \bigbiguplus A_2 \to D,$$

   both of them with the *same set* of two fused point-to-point arrows

   $$\frac{h_1}{\biguplus} = \frac{h_2}{\biguplus} = \left\{(j \circ f \circ k) \bigcup (m \circ g \circ k) : A_1 \to D, (j \circ f \circ l) \bigcup (m \circ g \circ l) : A_2 \to D\right\},$$

   so that $h_1 = h_2$, i. e., this composition of complex arrows is associative as well.

4. For $k : A \to B_1, l : A \to B_2, f : B_1 \to C, g : B_2 \to C, j : C \to D_1$ and $m : C \to D_2$, we consider the compositions $\langle j, m \rangle \circ [f, g] \circ \langle k, l \rangle : A \to B_1 \biguplus B_2 \to C \to D_1 \biguplus D_2$. Let us consider the two different orders of composition:

   4.1
   $$h_1 = \langle j, m \rangle \circ ([f, g] \circ \langle k, l \rangle) =^4 \langle j, m \rangle \circ \lceil f \circ k, g \circ l \rfloor$$
   $$=^7 \lceil \langle j \circ f \circ k, j \circ g \circ l \rangle, \langle m \circ f \circ k, m \circ g \circ l \rangle \rfloor : A \to D_1 \bigbiguplus D_2,$$

   4.2
   $$h_2 = (\langle j, m \rangle \circ [f, g]) \circ \langle k, l \rangle =^6 [\langle j \circ f, m \circ f \rangle, \langle j \circ g, m \circ g \rangle] \circ \langle k, l \rangle$$
   $$=^4 \lceil \langle j \circ f \circ k, m \circ f \circ k \rangle, \langle j \circ g \circ l, m \circ g \circ l \rangle \rfloor : A \to D_1 \bigbiguplus D_2.$$

   They are equal, $h_1 = h_2$, because

   $$\frac{h_1}{\biguplus} = \frac{h_2}{\biguplus} = \left\{(j \circ f \circ k) \bigcup (j \circ g \circ l) : A \to D_1, (m \circ f \circ k) \bigcup (m \circ g \circ l) : A \to D_2\right\}.$$

7.1.1 Morphism properties of DB category

From the examples above, we have seen that there are different structural representations for the same arrow between two complex objects. That is, for a given set of ptp arrows between two complex objects, we can have a number of different structural representations that are mutually equal. Thus, it is natural to fix a canonical representation of such equivalent classes of arrows and to use them whenever possible. The fact is that each arrow in **DB** can be substituted by such canonical arrow, in an analogous way as any object can be substituted by an isomorphic object.

Lemma 2. *Let us define the function* $\circledast : \mathrm{Mor}_{\mathrm{DB}} \to \mathrm{Mor}_{\mathrm{DB}}$*, such that for any arrow* $f : A \to B$ *where* $A = \biguplus_{1\le j\le m} A_j$ *and* $B = \biguplus_{1\le i\le k} B_i$*,* $m, k \ge 1$*,*

$$\circledast(f) \triangleq [\langle h_{11}, \ldots, h_{1k}\rangle, \ldots, \langle h_{m1}, \ldots, h_{mk}\rangle] : A \to B,$$

where $h_{ji} = f_{ji}$ *if* $(f_{ji} : A_j \to B_i) \in \frac{f}{\biguplus}$*;* $\perp^1$ *otherwise.*

This mapping is an identity for the simple arrows. For any complex arrow, it returns an equal arrow that contains the maximal number of simple arrows in its structural representation. For a given set $S = \frac{f}{\biguplus}$ *of ptp arrows between A and B, by* $\widehat{\circledast}(A, B, S) \triangleq \circledast(f)$ *we denote a canonical arrow obtained from S.*

Proof. There are the following cases:

1. The case when $m = k = 1$, so that $f : A_1 \to B_1$ is a simple arrow. Then $\frac{f}{\biguplus} = \{f\}$ if $\widetilde{f} \neq \perp^0$; $\emptyset$ otherwise.
   Consequently, $\circledast(f) = f$ if $\frac{f}{\biguplus} = \{f\}$, or $\circledast(f) = \perp^1$ if $\frac{f}{\biguplus} = \emptyset$, but from the fact that in this case $\widetilde{f} = \perp^0 = \widetilde{\perp^1}$, from Definition 44, we obtain again that $\circledast(f) = f$.
2. The case of $m \cdot k \ge 2$ (a complex arrow) and $\frac{f}{\biguplus} = \emptyset$. Let us consider the following cases:
   2.1 $m = 1$. Then $\circledast(f) = [\langle h_{11}, \ldots, h_{1k}\rangle] =$ (in $[_]$ we have only one element) $= \langle h_{11}, \ldots, h_{1k}\rangle$ where $h_{1i} = \perp^1$ for all $i = 1, \ldots, k$, so that $\frac{\circledast(f)}{\biguplus} = \emptyset = \frac{f}{\biguplus}$, and hence, $\circledast(f) = f$ is maximal with $1 \cdot k = k \ge 2$ simple arrows.
   2.2 $k = 1$. Then $\circledast(f) = [\langle h_{11}\rangle, \ldots, \langle h_{m1}\rangle] = [h_{11}, \ldots, h_{m1}]$ where $h_{j1} = \perp^1$ for all $j = 1, \ldots, m$, so that $\frac{\circledast(f)}{\biguplus} = \emptyset = \frac{f}{\biguplus}$, and hence, $\circledast(f) = f$ is maximal with $m \cdot 1 = m \ge 2$ simple arrows.
   2.3 Both $m, k \ge 2$. Then $\circledast(f) = [\langle h_{11}, \ldots, h_{1k}\rangle, \ldots, \langle h_{m1}, \ldots, h_{mk}\rangle]$ where $h_{ji} = \perp^1$ for all $j = 1, \ldots, m$ and $i = 1, \ldots, k$, so that $\frac{\circledast(f)}{\biguplus} = \emptyset = \frac{f}{\biguplus}$, and hence, $\circledast(f) = f$ is maximal $k \cdot m \ge 2$ simple arrows.
3. The case of $m \cdot k \ge 2$ (a complex arrow) and $\frac{f}{\biguplus} \neq \emptyset$. Then, as in case 2, we have three possible cases, and in each of them we obtain that $\frac{\circledast(f)}{\biguplus} = \frac{f}{\biguplus}$, and hence, $\circledast(f) = f$ is maximal $k \cdot m \ge 2$ simple arrows. □

It is easy to see that for any f, $\circledast$ inserts only the new empty arrows $\perp^1$ in this canonical structural representation where, for each simple object A_j in a A, we have a full cone of simple arrows $\langle h_{j1}, \ldots, h_{jk}\rangle : A_j \to B$ from A_j into each simple object in B_i for $i = 1, \ldots, k$. For example, consider $\circledast(k)$ for $k = k_{11} \biguplus [k_{2,2}, k_{3,2}] : A_1 \biguplus A_2 \biguplus A_3 \to B_1 \biguplus B_2$.

Moreover, we are able to construct a well-defined complex arrow between any two complex objects by specifying the set S of ptp arrows. That is useful also when the set S of ptp arrows is empty. For example, $\widehat{\circledast}(A_1 \biguplus A_2, B_1, \{f_{21}\}) = [\perp^1, f_{21}] : A_1 \biguplus A_2 \to B_1$, or

$$\widehat{\circledast}(A_1, B_1 \Big\biguplus B_2, \emptyset) = \langle \perp^1, \perp^1 \rangle : A_1 \to B_1 \Big\biguplus B_2.$$

We can show the following important isomorphisms for (also complex) instance-databases in **DB**.

Lemma 3. *For any $A, B, C \in \mathrm{Ob}_{\mathrm{DB}}$, we have the following isomorphisms:*

1. *For any object A, we have the isomorphism $\mathrm{is}_A : A \to TA$ with its inverse $\mathrm{is}_A^{-1} : TA \to A$ such that $\widetilde{\mathrm{is}_A} = \widetilde{\mathrm{is}_A^{-1}} = \widetilde{\mathrm{id}_A} = TA$. Thus, $A \simeq TA$.*
2. *(Commutativity) $A \biguplus B \simeq B \biguplus A$.*
3. *(Associativity) $A \biguplus (B \biguplus C) \simeq A \biguplus B \biguplus C \simeq (A \biguplus B) \biguplus C$.*
4. *(Composition with zero object) $C \biguplus \perp^0 \simeq C \simeq \perp^0 \biguplus C$.*

Proof. Claim 1. Let us consider a simple object (instance-database) A. The SOtgd Φ_A of the schema mappings, corresponding to the morphisms $\mathrm{id}_A : A \to A$, $\mathrm{is}_A : A \to TA$ and $\mathrm{is}_A^{-1} : TA \to A$, is equal to the InverseOperads$(\{1_{r_i} \in O(r_i, r_i) \mid r_i \in S_A\} \bigcup \{1_{r_\emptyset}\})$, i. e., to the logical formula $\bigwedge\{\forall \mathbf{x}_i (r_i(\mathbf{x}_i) \Rightarrow r_i(\mathbf{x}_i)) \mid r_i \in S_A\}$ where $A = \alpha^*(\mathcal{A})$ of a schema $\mathcal{A}$ and $\mathcal{A} \subseteq \widehat{\mathcal{A}}$ (thus, $A = \alpha^*(\mathcal{A}) \subseteq \alpha^*(\widehat{\mathcal{A}}) = TA$). Thus, $\mathrm{id}_A = \alpha^*(\mathrm{MakeOperads}(\{\Phi_A\})) : A \to A$, $\mathrm{is}_A = \alpha^*(\mathrm{MakeOperads}(\{\Phi_A\})) : A \to TA$, and $\mathrm{is}_A^{-1} = \alpha^*(\mathrm{MakeOperads}(\{\Phi\}_A)) : TA \to A$, are *atomic* morphisms (Definition 39). That is, these three morphisms are based on the same SOtgd Φ_A and, consequently, for each object A in **DB** ($\widetilde{\mathrm{id}_A} = TA$ is demonstrated by Theorem 10) that $\widetilde{\mathrm{is}_A} = \widetilde{\mathrm{is}_A^{-1}} = \widetilde{\mathrm{id}_A} = TA$. For a complex object, $A = \biguplus_{1 \le j \le m} A_j$, $m \ge 2$, $\mathrm{is}_A = \biguplus_{1 \le j \le m} \mathrm{is}_{A_j}$, $\mathrm{is}_A^{-1} = \biguplus_{1 \le j \le m} \mathrm{is}_{A_j}^{-1}$: In fact, $\mathrm{is}_A^{-1} \circ \mathrm{is}_A = \biguplus_{1 \le i \le k} \mathrm{is}_{A_i}^{-1} \circ \mathrm{is}_{A_i} = \biguplus_{1 \le i \le k} \mathrm{id}_{A_i} = \mathrm{id}_A$ (i. e., $\frac{\mathrm{id}_A}{\biguplus} = \frac{\biguplus_{1 \le i \le k} \mathrm{id}_{A_i}}{\biguplus} = \{\mathrm{id}_{Ai} : A_i \to A_i \mid 1 \le i \le k\}$). Analogously, $\mathrm{is}_{A_i} \circ \mathrm{is}_{A_i}^{-1} = \biguplus_{1 \le i \le k} \mathrm{id}_{TA_i} = \mathrm{id}_{TA}$. Hence, $\widetilde{\mathrm{is}_A} = \widetilde{\mathrm{is}_A^{-1}} = \widetilde{\mathrm{id}_A} = \biguplus_{1 \le j \le m} TA_j = TA$.

Claim 2 (Commutativity). For any two objects in **DB**, there is a commutativity isomorphism $A \biguplus B \simeq B \biguplus A$.

Let us show that this isomorphism is represented by the complex morphism

$$i_1 = \langle [\perp^1, \mathrm{id}_B], [\mathrm{id}_A, \perp^1] \rangle : A \Big\biguplus B \to B \Big\biguplus A$$

(where $[\perp^1, \mathrm{id}_B] : A \biguplus B \to B$ and $[\mathrm{id}_A, \perp^1] : A \biguplus B \to A$) and its inverse by $i_2 = i_1^{\mathrm{OP}} = [[\perp^1, \mathrm{id}_B]^{\mathrm{OP}}, [\mathrm{id}_A, \perp^1]^{\mathrm{OP}}] = [\langle \perp^1, \mathrm{id}_B \rangle, \langle \mathrm{id}_A, \perp^1 \rangle] : B \biguplus A \to A \biguplus B$.

In fact,

$$\begin{aligned} i_1 \circ i_2 &= \langle [\perp^1, \mathrm{id}_B], [\mathrm{id}_A, \perp^1] \rangle \circ [\langle \perp^1, \mathrm{id}_B \rangle, \langle \mathrm{id}_A, \perp^1 \rangle] \\ &= \langle [[\perp^1, \mathrm{id}_B] \circ \langle \perp^1, \mathrm{id}_B \rangle, [\perp^1, \mathrm{id}_B] \circ \langle \mathrm{id}_A, \perp^1 \rangle], \\ &\quad [[\mathrm{id}_A, \perp^1] \circ \langle \perp^1, \mathrm{id}_B \rangle, [\mathrm{id}_A, \perp^1] \circ \langle \mathrm{id}_A, \perp^1 \rangle] \rangle \\ &= \langle [\lceil \mathrm{id}_B \rceil, \lceil \perp^1 \rfloor], [\lceil \perp^1 \rfloor, \lceil \mathrm{id}_A \rfloor] \rangle = \langle [\mathrm{id}_B, \perp^1], [\perp^1, \mathrm{id}_A] \rangle. \end{aligned}$$

Thus, $\frac{i_1 \circ i_2}{\biguplus} = \{\mathrm{id}_A, \mathrm{id}_B\} = \frac{\mathrm{id}_B \biguplus \mathrm{id}_A}{\biguplus}$, and hence, by Definition 44 $i_1 \circ i_2 = \mathrm{id}_B \biguplus \mathrm{id}_A =$ (point 2 of Example 22) $= \mathrm{id}_{B \biguplus A} : B \biguplus A \to B \biguplus A$. Analogously,

$$\begin{aligned} i_2 \circ i_1 &= [\langle \perp^1, \mathrm{id}_B \rangle, \langle \mathrm{id}_A, \perp^1 \rangle] \circ \langle [\perp^1, \mathrm{id}_B], [\mathrm{id}_A, \perp^1] \rangle \\ &= \lceil \langle \perp^1, \mathrm{id}_B \rangle \circ [\perp^1, \mathrm{id}_B], \langle \mathrm{id}_A, \perp^1 \rangle \circ [\mathrm{id}_A, \perp^1] \rfloor \\ &= \lceil \langle \perp^1 \circ [\perp^1, \mathrm{id}_B], \mathrm{id}_B \circ [\perp^1, \mathrm{id}_B] \rangle, \langle \mathrm{id}_A, \circ [\mathrm{id}_A, \perp^1], \perp^1 \circ [\mathrm{id}_A, \perp^1] \rangle \rfloor \\ &= \lceil \langle \perp^1, [\perp^1, \mathrm{id}_B] \rangle, \langle [\mathrm{id}_A, \perp^1], \perp^1 \rangle \rfloor : A \bigl\lfloor\!\biguplus\!\bigr\rfloor B \to A \bigl\lfloor\!\biguplus\!\bigr\rfloor B. \end{aligned}$$

Thus, $\frac{i_2 \circ i_1}{\biguplus} = \{\mathrm{id}_A, \mathrm{id}_B\} = \frac{\mathrm{id}_A \biguplus \mathrm{id}_B}{\biguplus}$, and hence, we obtain an identity $i_2 \circ i_1 = \mathrm{id}_A \biguplus \mathrm{id}_B = \mathrm{id}_{A \biguplus B} : A \biguplus B \to A \biguplus B$. Consequently, $i_1 = \langle [\perp^1, \mathrm{id}_B], [\mathrm{id}_A, \perp^1] \rangle : A \biguplus B \to B \biguplus A$ is an isomorphism, i. e., $A \biguplus B \simeq B \biguplus A$.

Claim 3 (Associativity). It holds that $\mathrm{id}_A \biguplus \langle [\mathrm{id}_B, \perp^1], [\perp^1, \mathrm{id}_C] \rangle : A \biguplus (B \biguplus C) \to A \biguplus B \biguplus C$ is an isomorphism. It is enough to show that $i_3 = \langle [\mathrm{id}_B, \perp^1], [\perp^1, \mathrm{id}_C] \rangle : B \biguplus C \to B \biguplus C$ is an isomorphism, as follows: $\frac{i_3}{\biguplus} = \{\mathrm{id}_B, \mathrm{id}_C\} = \frac{\mathrm{id}_{B \biguplus C}}{\biguplus}$ (from the fact that $\mathrm{id}_{B \biguplus C} = \mathrm{id}_B \biguplus \mathrm{id}_C$), thus (by point 3 of Definition 44) $i_3 = \mathrm{id}_{B \biguplus C} : B \biguplus C \to B \biguplus C$ and from the fact that each identity morphism is an isomorphism, it holds that i_3 is an isomorphism.

Claim 4. $T(C \bigl\lfloor\!\biguplus\!\bigr\rfloor \perp^0) = TC \bigl\lfloor\!\biguplus\!\bigr\rfloor T \perp^0 = TC \bigl\lfloor\!\biguplus\!\bigr\rfloor \perp^0 \neq TC$, and hence, we cannot use Proposition 25 in order to demonstrate the isomorphism $C \bigl\lfloor\!\biguplus\!\bigr\rfloor \perp^0 \simeq C$, but the standard categorial method for verification of the isomorphisms. For a given schema $\mathcal{C}$ with the interpretation α such that $C = \alpha^*(\mathcal{C})$, we have the identity morphism $\mathrm{id}_C = \alpha^*(\mathbf{M}_{CC}) : C \to C$ with $\mathbf{M}_{CC} = \alpha^*(\{1_r \mid r \in \mathcal{C}\} \bigcup \{1_{r_\emptyset}\})$.

Let us define the morphisms $i_1 = \langle \alpha^*(\mathbf{M}_{CC}), \alpha^*(\{1_{r_\emptyset}\}) \rangle : C \to C \bigl\lfloor\!\biguplus\!\bigr\rfloor \perp^0$ and $i_2 = [\alpha^*(\mathbf{M}_{CC}), \alpha^*(\{1_{r_\emptyset}\})] : C \bigl\lfloor\!\biguplus\!\bigr\rfloor \perp^0 \to C$. Thus,

$$\begin{aligned} i_2 \circ i_1 &= [\alpha^*(\mathbf{M}_{CC}), \alpha^*(\{1_{r_\emptyset}\})] \circ \langle \alpha^*(\mathbf{M}_{CC}), \alpha^*(\{1_{r_\emptyset}\}) \rangle = [\mathrm{id}_C, \perp^1] \circ \langle \mathrm{id}_C, \perp^1 \rangle \\ &= \lceil \mathrm{id}_C \circ \mathrm{id}_C, \perp^1 \circ \perp^1 \rfloor = \lceil \mathrm{id}_C, \perp^1 \rfloor = \text{(from point 2.4 of Definition 41)} \\ &= \mathrm{id}_C : C \to C, \end{aligned}$$

i. e., the identity morphisms for C. While,

$$i_1 \circ i_2 = \langle \alpha^*(\mathbf{M}_{CC}), \alpha^*(\{1_{r_\emptyset}\}) \rangle \circ [\alpha^*(\mathbf{M}_{CC}), \alpha^*(\{1_{r_\emptyset}\})]$$

$$= \langle \mathrm{id}_C, \perp^1 \rangle \circ [\mathrm{id}_C, \perp^1] = \langle [\mathrm{id}_c, \perp^1], [\perp^1, \perp^1] \rangle.$$

Thus, $\frac{i_1 \circ i_2}{\biguplus} = \frac{\mathrm{id}_{C \biguplus \perp^0}}{\biguplus} = \{\mathrm{id}_C\}$ (from $\mathrm{id}_{C \biguplus \perp^0} = \mathrm{id}_C \biguplus \perp^1$), and hence, from point 3 of Definition 44, $i_1 \circ i_2 = \mathrm{id}_{C \biguplus \perp^0}$. Consequently, this morphism is the identity morphism of the object $C \biguplus \perp^0$. Therefore, both morphisms $i_1 \circ i_2$ and $i_2 \circ i_1$ are the identity morphisms and so $i_1 = \langle \mathrm{id}_C, \perp^1 \rangle : C \to C \biguplus \perp^0$ is an isomorphism, and hence,
(a) $C \simeq C \biguplus \perp^0$ and, analogously, by commutativity, $\perp^0 \biguplus C \simeq C \biguplus \perp^0 \simeq C$.

Let us show that i_1 is monic and epic as well. For any two morphisms, $g_1, g_2 : D \to C$ such that $i_1 \circ g_1 = i_1 \circ g_2$, i. e., $i_1 \circ g_1 = \langle \mathrm{id}_C, \perp^1 \rangle \circ g_1 = \langle g_1, \perp^1 \rangle = i_1 \circ g_2 = \langle g_2, \perp^1 \rangle$, we obtain $g_1 = g_2$ (from $\{g_1\} = \frac{i_1 \circ g_1}{\biguplus} = \frac{i_1 \circ g_2}{\biguplus} = \{g_2\}$ and by point 3 of Definition 44), and hence, i_1 is a monomorphism. For any two, $g_1 = [k_1, \perp^1]$ and $g_2 = [k_2, \perp^1] : C + \perp^0 \to D$ such that $g_1 \circ i_1 = g_2 \circ g_1$, i. e., $g_1 \circ i_1 = [k_1, \perp^1] \circ \langle \mathrm{id}_C, \perp^1 \rangle = \lceil k_1, \perp^1 \rfloor = k_1$ (component $\perp^1$ is eliminated according to point 4 in Definition 41) $= g_2 \circ i_1 = [k_2, \perp^1] \circ \langle \mathrm{id}_C, \perp^1 \rangle = \lceil k_2, \perp^1 \rfloor = k_2$, we obtain $k_1 = k_2$, thus $g_1 = g_2$, and hence, i_1 is an epimorphism. □

Remark. From the fact that $C \biguplus \perp^0$ is isomorphic to C and that two isomorphic objects in a category can be substituted, we can eliminate for complex objects all their components, which are equal to the empty database $\perp^0$, and generally, consider only the *strictly-complex* objects. Now we obtain the first fundamental property of **DB** category, which confirms that it is fundamentally different from **Set**.

Proposition 22. *The morphisms* $[\mathrm{id}_C, \mathrm{id}_C] : C \biguplus C \to C$ *are epic and monic but not isomorphic. Consequently,* **DB** *is not a topos.*

Proof. Let for any two morphisms $g_1 = \langle h_1, k_1 \rangle : C \to C \biguplus C$ and $g_2 = \langle h_2, k_2 \rangle : C \to C \biguplus C$, $[\mathrm{id}_C, \mathrm{id}_C] \circ g_1 = [\mathrm{id}_C, \mathrm{id}_C] \circ g_2 : C \to C$. Let us show that $g_1 = g_2$. In fact, $[\mathrm{id}_C, \mathrm{id}_C] \circ g_1 = [\mathrm{id}_C, \mathrm{id}_C] \circ \langle h_1, k_1 \rangle = \lceil \mathrm{id}_C \circ h_1, \mathrm{id}_C \circ k_1 \rfloor = \lceil h_1, k_1 \rfloor : C \to C$ and $[\mathrm{id}_C, \mathrm{id}_C] \circ g_2 = \lceil h_2, k_2 \rfloor : C \to C$. Consequently, $\lceil h_1, k_1 \rfloor = \lceil h_2, k_2 \rfloor$ and, from Definition 44, there exists a bijection σ such that $\sigma(h_1) = h_2, \sigma(k_1) = k_2$ with $\widetilde{h_1} = \widetilde{\sigma(h_1)} = \widetilde{h_2}$ and $\widetilde{k_1} = \widetilde{\sigma(k_1)} = \widetilde{k_2}$, so that $h_1 = h_2$ and $k_1 = k_2$, and hence, $g_1 = \langle h_1, k_1 \rangle = \langle h_2, k_2 \rangle = g_2$. Consequently, $[\mathrm{id}_C, \mathrm{id}_C] : C \biguplus C \to C$ is a monomorphism.

Let for any two morphisms $g_1 : C \to C$ and $g_2 : C \to C$, $g_1 \circ [\mathrm{id}_C, \mathrm{id}_C] = g_2 \circ [\mathrm{id}_C, \mathrm{id}_C]$ (the case when $g_1 = g_2 = \perp^1$ is trivial, and hence, we consider the cases when g_1 and g_2 are different from $\perp^1$). Let us show that $g_1 = g_2$. In fact, $g_1 \circ [\mathrm{id}_C, \mathrm{id}_C] = [g_1, g_1] : C \biguplus C \to C$ and $[g_2 \circ \mathrm{id}_C, \mathrm{id}_C] = [g_2, g_2] : C \to C$. Consequently, $[g_1, g_1] = [g_2, g_2]$ and, from point 3 of Definition 44, $g_1 = g_2$. Consequently, $[\mathrm{id}_C, \mathrm{id}_C] : C \biguplus C \to C$ is an epimorphism. Let us show that $f = [\mathrm{id}_C, \mathrm{id}_C] : C \biguplus C \to C$ is not an isomorphism: If we suppose that it is an isomorphism, then $f \circ f^{-1} : C \to C$ has to be the identity arrow id_C. From the fact that $f^{-1} = [\mathrm{id}_C, \mathrm{id}_C]^{\mathrm{OP}} = \langle \mathrm{id}_C^{\mathrm{OP}}, \mathrm{id}_C^{\mathrm{OP}} \rangle = \langle \mathrm{id}_C, \mathrm{id}_C \rangle : C \to C \biguplus C$, so that $f \circ f^{-1} = [\mathrm{id}_C, \mathrm{id}_C] \circ \langle \mathrm{id}_C, \mathrm{id}_C \rangle = \lceil \mathrm{id}_C \circ \mathrm{id}_C, \mathrm{id}_C \circ \mathrm{id}_C \rfloor = \lceil \mathrm{id}_C, \mathrm{id}_C \rfloor \neq \mathrm{id}_C$ (by point 2 of Definition 44), i. e., the properties of an isomorphism are not satisfied.

From the fact that in any topos category if a morphism is both epic and monic, then it must also be an isomorphism, we conclude that **DB** is not a topos. □

Thus, **DB** is not a topos. However, there is a subset of morphisms for which the topological properties hold so that if a morphism is both epic and monic then it must also be an isomorphism: it is the set of all simple arrows, as we will show in next corollary, so that first step is to analyze the topological properties of the simple morphisms in **DB** and only after that the general properties for (also complex) morphisms.

Corollary 17. *For each simple arrow $f : A \to B$, the following are valid:*
1. *f is monic iff $\widetilde{f} = TA$.*
2. *f is epic iff $\widetilde{f} = TB$.*
3. *f is an isomorphism iff it is monic and epic (i. e., when $\widetilde{f} = TA = TB$).*

Proof. Claim 1 (from right to left). Let us show that if $\widetilde{f} = TA$ then $f : A \to B$ is monic. A simple arrow $f : A \longrightarrow B$ is monic iff for any (possibly complex) two arrows $h, g : \biguplus_{1\le i\le n} X_i \longrightarrow A, f\circ h = f\circ g$ implies $h = g$. Let us show that it is satisfied when $\widetilde{f} = TA$. We have (from Proposition 23) that for each point-to-point simple arrow $(g_i : X_i \to A) \in \frac{g}{\biguplus}$, $\widetilde{g_i} \subseteq TA$ (analogously for each $(h_i : X_i \to A) \in \frac{h}{\biguplus}$, $\widetilde{h_i} \subseteq TA$). Thus, if $f\circ h = f\circ g$ then, from Definition 43, $\frac{f\circ g}{\biguplus} = \frac{f\circ h}{\biguplus}$, i. e., they have the same ptp arrows, i. e., $f\circ g_i = f\circ h_i : X_i \to B$, so that they have the same fluxes, i. e., $\widetilde{f\circ g_i} = \widetilde{f\circ h_i}$. But we have that $\widetilde{f\circ g_i} = \widetilde{f}\bigcap\widetilde{g}_i = TA\bigcap\widetilde{g}_i = \widetilde{g}_i$ and $\widetilde{f\circ h_i} = \widetilde{f}\bigcap\widetilde{h}_i = TA\bigcap\widetilde{h}_i = \widetilde{h}_i$, so that $\widetilde{h}_i = \widetilde{g}_i$, i. e., are equal ptp arrows $h_i = g_i : X_i \to A$. It holds for all ptp arrows in g and h, so that $g = h$, and, consequently, $f : A \longrightarrow B$ is a monomorphism.

(From left to right). If a simple arrow $f : A \hookrightarrow B$ is monic, then for any (possibly complex) two arrows $h, g : \biguplus_{1\le i\le n} X_i \longrightarrow A, f\circ h = [f\circ h_1, \dots, f\circ h_n] = f\circ g = [f\circ g_1, \dots, f\circ g_n] : \biguplus_{1\le i\le n} X_i \longrightarrow B$, implies $h = g$, i. e., $\forall 1 \le l \le n (f\circ g_l = f\circ h_l : X_l \to B$ implies $g_l = h_l : X_l \to A)$. Let us suppose that $\widetilde{f} \subset TA$, so that we can have, e. g., $\widetilde{g_l} = \widetilde{f} \neq TA = \widetilde{h_l}$, i. e., $g_l \neq h_l$ but with $\widetilde{f\circ g_l} = \widetilde{f} = \widetilde{f\circ h_l}$, i. e., $f\circ g_l = f\circ h_l$. What is a contradiction? Thus, it must hold that $\widetilde{f} = TA$.

Claim 2. The proof is analog to that of Claim 1.

Claim 3 (from left to right). It holds for every category.

(From right to left). If $f : A \to B$ is both monic and epic, then from the first two points, $\widetilde{f} = TA = TB$. Let us show that $f = \mathrm{is}_B^{-1}\circ \mathrm{is}_A$. In fact, $\widetilde{\mathrm{is}_B^{-1}\circ \mathrm{is}_A} = \widetilde{\mathrm{is}_B^{-1}}\bigcap\widetilde{\mathrm{is}_A} =$ (from Lemma 3) $= TB\bigcap TA = TA = TB$, so that from Definition 44, $f = \mathrm{is}_B^{-1}\circ \mathrm{is}_A : A \to B$, i. e., f is a composition of two isomorphisms, and hence it is an isomorphism as well. □

Now we will also analyze the properties of complex arrows.

Notice that the complex morphism $f = [\mathrm{id}_C, \mathrm{id}_C] : C \biguplus C \to C$ has to be formally represented by "indexing by position" formalism, i. e., $f = [\mathrm{id}_C, \mathrm{id}_C] : A_1 \biguplus A_2 \to B_1$ where $A_1 = A_2 = B_1 = C$, so that $\frac{f}{\biguplus} = \{f_{11} = \mathrm{id}_C : A_1 \to B_1, f_{21} = \mathrm{id}_C : A_2 \to B_1\}$, i. e., by *two* mutually independent mapping morphisms $\mathrm{id}_C : C \to C$. Thus, the morphisms

$g = [\mathrm{id}_C, \perp^1] : C \uplus C \to C$ and $h = [\perp^1, \mathrm{id}_C] : C \uplus C \to C$ have $\frac{g}{\uplus} = \{g_{11} = \mathrm{id}_C : A_1 \to B_1\} = \{\mathrm{id}_C : C \to C\} \neq \frac{f}{\uplus}, \frac{h}{\uplus} = \{h_{21} = \mathrm{id}_C : A_2 \to B_1\} = \{\mathrm{id}_C : C \to C\} \neq \frac{f}{\uplus}$. But $\frac{g}{\uplus} = \{\mathrm{id}_C : C \to C\} = \frac{h}{\uplus}$ so that, from Definition 44, $g = h$. Thus, $[\mathrm{id}_C, \perp^1] = g = h = [\perp^1, \mathrm{id}_C]$.

Proposition 23. *We have the following fundamental properties for the morphisms in* **DB***:*

1. *if a morphism $f : A \longrightarrow B$ has a property $P \in$ {isomorphic, monic, epic}, then each of its ptp arrow $f_l \in \frac{f}{\uplus}$ has the same property P;*
2. *if $f : A \hookrightarrow B$ is monic and $A = \biguplus_{1 \leq j \leq m} A_j$, $m \geq 1$, is strictly complex then for each A_j it must exist with a ptp arrow $f_n \in \frac{f}{\uplus}$ with $\mathrm{dom}(f_n) = A_j$;*
3. *if $f : A \twoheadrightarrow B$ is epic and $B = \biguplus_{1 \leq i \leq k} B_i$, $k \geq 1$, $k \geq 1$, is strictly complex then for each B_i there must exist a ptp arrow $f_n \in \frac{f}{\uplus}$ with $\mathrm{cod}(f_n) = B_i$;*
4. *if $C = \biguplus_{1 \leq j \leq m} C_j$ and $B = \biguplus_{1 \leq i \leq k} B_i$ are strictly-complex objects and there is an isomorphism* is $: C \simeq B$, *then $m = k$.*

Proof. 1. Its inverted arrow is $\frac{f^{\mathrm{OP}}}{\uplus} = \{f_{ji}^{\mathrm{OP}} : B_i \to A_j \mid f_{ji} : A_j \to B_i \text{ in } \frac{f}{\uplus} \text{ and } \widetilde{f_{ji}^{\mathrm{OP}}} = \widetilde{f_{ji}}\}$. Let $f : \biguplus_{1 \leq j \leq m} A_j \to \biguplus_{1 \leq i \leq k} B_i$ be an isomorphism, so that $f \circ f^{-1} = \mathrm{id}_B$, thus $\frac{f \circ f^{-1}}{\uplus} = \frac{f \circ f^{\mathrm{OP}}}{\uplus} = \{f_{ji} \circ f_{ji}^{\mathrm{OP}} = \mathrm{id}_i : B_i \to B_i \mid f_{ji} \in \frac{f}{\uplus}, f_{ji}^{\mathrm{OP}} \in \frac{f^{\mathrm{OP}}}{\uplus}$ with $\mathrm{cod}(f_{ji}^{\mathrm{OP}}) = \mathrm{dom}(f_{ji}) = A_j$ and $\mathrm{cod}(f_{ji}) = B_i\} = \frac{\biguplus_{1 \leq i \leq k} \mathrm{id}_{B_i}}{\uplus} = \frac{\mathrm{id}_B}{\uplus}$. Let us suppose that $f_{ji} \in \frac{f}{\uplus}$ is not an isomorphism. Thus, $f_{ji} \circ f_{ji}^{\mathrm{OP}} \neq \mathrm{id}_i : B_i \to B_i$ (with $\mathrm{cod}(f_{ji}^{\mathrm{OP}}) = \mathrm{dom}(f_{ji})$ and $\mathrm{cod}(f_{ji}) = B_i$), so that $f \circ f^{-1} \neq \mathrm{id}_B$ and f would not be an isomorphism, which is a contradiction. Consequently, each $f_{ji} \in \frac{f}{\uplus}$ must be an isomorphism. The converse does not hold: let us consider the morphism $f = [\mathrm{id}_C, \mathrm{id}_C] : A_1 \uplus A_2 \to B_1$, with $A_1 = A_2 = B_1 = C$ where $\frac{f}{\uplus} = \{f_{11} = \mathrm{id}_C : A_1 \to B_1, f_{21} = \mathrm{id}_C : A_2 \to B_1\}$ where the point-to-point arrows id_C are the identity arrows (thus isomorphisms as well) but f is not an isomorphism (from Proposition 22).

Let $f : A \hookrightarrow B$ be a monomorphism. Then, for every two arrows $g, h : X \to A$ if $f \circ g = f \circ h$, then we obtain $g = h$. Let us suppose that there exists $(f_{ji} : A_j \to B_i) \in \frac{f}{\uplus}$, which is not monic, i. e., there exist g_{lj}, h_{lj} such that $f_{ji} \circ g_{lj} = f_{ji} \circ h_{lj}$ with $g_{lj} \neq h_{lj} : X_l \to A_j$. However, from the fact that f is a monomorphism it must hold that $g = h$ and, by Definition 44, $\frac{g}{\uplus} = \frac{h}{\uplus}$ with $g_{lj} = h_{lj} : X_l \to A_j$, which is a contradiction. Thus, each point-to-point arrow f_{ji} of a monic arrow f must be monic as well.

The proof for an epimorphism $f : A \twoheadrightarrow B$ is similar.

Claim 2. Let A be a strictly-complex object (with all $A_j \neq \perp^0$, $j = 1, \ldots, m$) and let us suppose that there exists A_j such that $A_j \notin \{\mathrm{dom}(f_n) \mid f_n \in \frac{f}{\uplus}\}$. Let $X = X_1$ be a simple object and $g, h : X_1 \to A$ be two arrows such that $f \circ g = f \circ h$ so that from the fact that f is a monomorphism, $g = h$ with two ptp arrows $(g_{1j} : X_1 \to A_j) \in \frac{g}{\uplus}$ and $(h_{1j} : X_1 \to A_j) \in \frac{h}{\uplus}$ with $g_{1j} = h_{1j}$. However, we can now take any ptp arrow $g'_{1j} : X_1 \to A_j$ such that $g'_{1j} \neq h_{1j}$ and change g into g' by substituting old g_{1j} by g'_{1j}, so that still $f \circ g' = f \circ h$ is valid, i. e., $f \circ g' = f \circ g$ but not $g' = g$, and hence, we obtain that f is not monomorphic, which is a

contradiction. Consequently, for each A_j there must exist at least one ptp arrow $f_n \in \frac{f}{\biguplus}$ with $\mathrm{dom}(f_n) = A_j$ if f is a monomorphism (note that $f_n \neq \bot^1$).

Claim 3. The proof when f is epic is similar.

Claim 4. Let for any two strictly-complex objects $C = \biguplus_{1\leq j\leq m} C_j$ and $B = \biguplus_{1\leq i\leq k} B_i$, $\mathrm{is} : C \to B$ be an isomorphism. Thus, $\{\mathrm{dom}(\mathrm{is}_i) \mid \mathrm{is}_i \in \frac{\mathrm{is}}{\biguplus}\} = \{C_1, \ldots, C_m\}$ and $\{\mathrm{cod}(\mathrm{is}_i) \mid \mathrm{is}_i \in \frac{\mathrm{is}}{\biguplus}\} = \{B_1, \ldots, B_k\}$ (from the fact that $\mathrm{is} \circ \mathrm{is}_1 = \mathrm{id}_B = \biguplus_{1\leq i\leq k} \mathrm{id}_{B_i}$ and $\mathrm{is}^{-1} \circ \mathrm{is} = \mathrm{id}_A = \biguplus_{1\leq j\leq m} \mathrm{id}_{A_j}$). Then, from point 1 of this proposition, each point-to-point arrow $\mathrm{is}_i \in \frac{\mathrm{is}}{\biguplus}$ is an isomorphism with $|\frac{\mathrm{is}}{\biguplus}| \geq \max(m, k)$. Let us show that $m = k$. Let us suppose that $m > k$, so that we have two different isomorphisms $\mathrm{is}_i, \mathrm{is}_j \in \frac{\mathrm{is}}{\biguplus}$ with $\mathrm{dom}(\mathrm{is}_i) \neq \mathrm{dom}(\mathrm{is}_j)$ and $\mathrm{cod}(\mathrm{is}_i) = \mathrm{cod}(\mathrm{is}_j) = B_l$. Thus, $\langle \mathrm{is}_i, \mathrm{is}_j \rangle \circ [\mathrm{is}_i^{\mathrm{OP}}, \mathrm{is}_j^{\mathrm{OP}}] = \lceil \mathrm{is}_i \circ \mathrm{is}_i^{\mathrm{OP}}, \mathrm{is}_j \circ \mathrm{is}_j^{\mathrm{OP}} \rfloor = \lceil \mathrm{id}_{B_l}, \mathrm{id}_{B_l} \rfloor \neq \mathrm{id}_{B_l}$. Hence, is cannot be an isomorphism, which is a contradiction. Analogously, if we suppose that $m < k$ we obtain again a contradiction. Consequently, it must be $m = k$. □

Remark. Claim 2 of this proposition explains the fact that when $C \neq \bot^0$, the morphism $k = [\mathrm{id}_c, \mathrm{id}_c] : C \biguplus C \to C$ where both point-to-point arrows are identities (thus, the isomorphisms) is not an isomorphism (but only a monic and epic arrow, as established by Proposition 22). While, for nonstrictly-complex objects, we have the isomorphisms when $m \neq k$ as well: let us consider the isomorphism $C \biguplus \bot^0 \simeq C$, when $m = 2$ ($A_1 = C$, $A_2 = \bot^0$) and $k = 1$ with $B_1 = C$.

We obtain the following important results for the *monomorphisms* in **DB**.

Proposition 24. *The following monomorphisms in the* **DB** *category are derivable from the morphisms and from the inclusions of the objects:*

1. *The conceptualized object $\widetilde{f} = B_\top(f)$, obtained from a morphism $f : A \longrightarrow B$, is a closed object in* **DB** *such that $\widetilde{f} = T\widetilde{f} \preceq A \otimes B$. This PO relation may be represented internally in* **DB** *by the monomorphism $\mathrm{in} : \widetilde{f} \hookrightarrow \widetilde{\mathrm{id}_A} \otimes \widetilde{\mathrm{id}_B}$ and its derivations $\mathrm{in}_A : \widetilde{f} \hookrightarrow \widetilde{\mathrm{id}_A}$ and $\mathrm{in}_B : \widetilde{f} \hookrightarrow \widetilde{\mathrm{id}_B}$ where $\widetilde{\mathrm{id}_A} = TA = \biguplus_{1\leq j\leq m} TA_j$, $m \geq 1$ and $\widetilde{\mathrm{id}_B} = TB = \biguplus_{1\leq i\leq k} TB_i$, $k \geq 1$, with ptp arrows:*

$$\left\{\mathrm{in}_{ji} = \{\mathrm{id}_R \mid R \in \widetilde{f_{ji}}\} : \widetilde{f_{ji}} \hookrightarrow S_{ji} \mid (f_{ji} : A_j \to B_i) \in \frac{f}{\biguplus}\right\},$$

   where $S_{ji} = TA_j \bigcap TB_i$ for $\frac{in}{\biguplus}$, $S_{ji} = TA_j$ for $\frac{\mathrm{in}_A}{\biguplus}$, and $S_{ji} = TB_i$ for $\frac{\mathrm{in}_B}{\biguplus}$.
2. *For any two objects $TA \subseteq TB$, there exists a monomorphism $\mathrm{in}_{TA} : TA \hookrightarrow TB$.*

Proof. Claim 1. Each conceptualized object (information flux) of an atomic arrow is a closed object. In fact, from Definition 67, for a simple atomic morphism $f = \alpha^*(\mathbf{M}_{AB}) : A \to B$, both databases A and B are simple objects (sets of relations) and so are TA and TB. Thus, if $\widetilde{f} = \bot^0$ (empty flux), then $T \bot^0 = \bot^0$. Otherwise, if $\widetilde{f} \neq \bot^0$ then: $T\widetilde{f} = T(T\{\pi_{\mathbf{x}_i} \| e[(_)_j / r_{i,j}]_{1\leq j\leq k} \|_A \mid q_i \in O(r_{i,1}, \ldots, r_{i,k}, r_i')$ is equal to $(e \Rightarrow (_)(\mathbf{t}_i)) \in \mathbf{M}_{AB}\}) = T\{\pi_{\mathbf{x}_i} \| e[(_)_j / r_{i,j}]_{1\leq j\leq k} \|_A \mid q_i \in O(r_{i,1}, \ldots, r_{i,k}, r_i')$ is equal to $(e \Rightarrow (_)(\mathbf{t}_i)) \in \mathbf{M}_{AB}\} = \widetilde{f} \subseteq TA$. From the fact that this information flux is transferred into B, $\widetilde{f} \subseteq TB$, so that $\widetilde{f} \subseteq$

$TA \bigcap TB$, i. e., by Definition 40, $\widetilde{f} \preceq A \otimes B$. Each arrow is a composition of a number of atomic arrows and the intersection of closed objects (their information fluxes) is always a closed object.

If $f : \biguplus_{1\leq j\leq m} A_j \to \biguplus_{1\leq i\leq k} B_i$ is a complex morphism with a set of more than one point-to-point arrow in $\frac{f}{\biguplus}$ then, for each $f_{ji} : A_j \to B_i \in \frac{f}{\biguplus}$, $\widetilde{f_{ji}} \subseteq TA_j \bigcap TB_i$. Hence, $\widetilde{\biguplus \frac{f}{\biguplus}} \preceq A \otimes B$ and $\widetilde{f} \preceq \widetilde{\biguplus \frac{f}{\biguplus}}$ (from Definition 43, $\widetilde{f} \simeq \widetilde{\biguplus \frac{f}{\biguplus}}$), so that $\widetilde{f} \preceq A \otimes B$. From Lemma 3, $\widetilde{\mathrm{id}_A} = TA$ and, from a Definition 40 and $T(TA) = TA$, $A \otimes B = TA \otimes TB = \widetilde{\mathrm{id}_A} \otimes \widetilde{\mathrm{id}_B}$. Thus, we obtain the PO relation $\widetilde{f} \preceq \widetilde{\mathrm{id}_A} \otimes \widetilde{\mathrm{id}_B}$ and we will show that it can be internally represented in **DB** by a monomorphism $\mathrm{in} : \widetilde{f} \hookrightarrow \widetilde{\mathrm{id}_A} \otimes \widetilde{\mathrm{id}_B}$. The facts that in, in_A and in_B are the morphisms defined by Theorem 10, and hence, they can be obtained from particular schema mappings and a given R-algebra α, will be shown in Theorem 14.

Let us consider a morphism $f : \biguplus_{1\leq j\leq m} A_j \to \biguplus_{1\leq i\leq k} B_i$, $k, m \geq 1$, with point-to-point (simple) arrows $(f_{ji} : A_j \to B_i) \in \frac{f}{\biguplus}$. Let us define the morphism $\mathrm{in} : \widetilde{f} \to TA \otimes TB$ with $\frac{in}{\biguplus} = \{\mathrm{in}_{ji} = \{\mathrm{id}_R : R \to R \mid R \in \widetilde{f_{ji}}\} : \widetilde{f_{ji}} \hookrightarrow TA_j \bigcap TB_i \mid (f_{ji} : A_j \to B_i) \in \frac{f}{\biguplus}\}$. Thus, there is a bijection $\sigma : \frac{f}{\biguplus} \to \frac{in}{\biguplus}$ with $\mathrm{in}_{ji} = \sigma(f_{ji}) : \widetilde{f_{ji}} \to TA_j \bigcap TB_i$ and $\widetilde{\mathrm{in}_{jk}} = \widetilde{f_{jk}}$ for any ptp arrow, and hence, $\widetilde{in} \simeq \widetilde{\biguplus \frac{in}{\biguplus}} \simeq \widetilde{\biguplus \frac{f}{\biguplus}} \simeq \widetilde{f}$.

For each pair $g, h : \biguplus_{1\leq l\leq n} X_l \to \widetilde{f}$ of morphisms (with $(g_{i,ji} : X_l \to \widetilde{f_{ji}}) \in \frac{g}{\biguplus} \neq \emptyset$ and $(h_{l,ji} : X_l \to \widetilde{f_{ji}}) \in \frac{h}{\biguplus} \neq \emptyset$ so that, from Proposition 23, $\widetilde{g_{l,ji}} \subseteq T\widetilde{f_{ji}} = \widetilde{f_{ji}}$ and $\widetilde{h_{l,ji}} \subseteq T\widetilde{f_{ji}} = \widetilde{f_{ji}}$, and hence, $\sigma(f_{ji}) \circ g_{l,ji} \in \frac{in \circ g}{\biguplus}$ and $\sigma(f_{ji}) \circ h_{l,ji} \in \frac{in \circ h}{\biguplus}$) such that $in \circ g = in \circ h : X \to B$, i. e., from point 3 of Definition 44, $\frac{in \circ g}{\biguplus} = \frac{in \circ h}{\biguplus}$. Thus, they have the same ptp arrows, i. e., $\sigma(f_{ji}) \circ g_{l,ji} = \sigma(f_{ji}) \circ h_{l,ji} : X_l \to TA_j \bigcap TB_i$ so that they have the same fluxes, i. e., $\widetilde{\sigma(f_{ji}) \circ g_{l,ji}} = \widetilde{\sigma(f_{ji}) \circ h_{l,ji}}$. However, $\widetilde{\sigma(f_{ji}) \circ g_{l,ji}} = \widetilde{\sigma(f_{ji})} \bigcap \widetilde{g_{l,ji}} = \widetilde{f_{ji}} \bigcap \widetilde{g_{l,ji}} = \widetilde{g_{l,ji}}$ and, analogously, $\widetilde{\sigma(f_{ji})} \bigcap \widetilde{h_{l,ji}} = \widetilde{f_{ji}} \bigcap \widetilde{h_{l,ji}} = \widetilde{h_{l,ji}}$ so that, $\widetilde{h_{l,ji}} = \widetilde{g_{l,ji}}$, i. e., are equal ptp arrows $h_{l,ji} = g_{l,ji} : X_l \to \widetilde{f_{ji}}$. This holds for all ptp arrows in g and h so that $g = h$. That is, for each g, h such that $in \circ g = in \circ h : X \to B$, we obtained that $g = h$, so that:

(1) $\mathrm{in} : \widetilde{f} \hookrightarrow \widetilde{\mathrm{id}_A} \bigcap \widetilde{\mathrm{id}_B}$ is a monomorphism and, analogously,

(2) $\mathrm{in}_A : \widetilde{f} \hookrightarrow \widetilde{\mathrm{id}_A} = TA$ and $\mathrm{in}_B : \widetilde{f} \hookrightarrow \widetilde{\mathrm{id}_B} = TB$.

Claim 2. Let us show that for any two simple objects A and B, $TA \subseteq TB$ implies the existence of a monomorphism $\mathrm{in}_{TA} : TA \hookrightarrow TB$:

If $TA \subseteq TB$, then we have a morphism $f = \{\mathrm{id}_R : R \to R \mid R \in TA\} : TA \hookrightarrow TB$ with $\widetilde{f} = TA$. Then, from (2), we have a monomorphism $\mathrm{in}_{TA} : \widetilde{f} \hookrightarrow \widetilde{\mathrm{id}_{TB}} = T(TB) = TB$ so, by substituting $\widetilde{f}$ by TA, we obtain $\mathrm{in}_{TA} : TA \hookrightarrow TB$.

Notice that for any two complex objects A and B such that $TA \subseteq TB$, these complex objects must have the *same* number $k \geq 2$ of simple databases, i. e., it must be $A = \biguplus_{1\leq i\leq k} A_i$ and $B = \biguplus_{1\leq i\leq k} B_i$ with $A_i \subseteq B_i$, for $1 \leq i \leq k$, and $TA = \biguplus_{1\leq i\leq k} TA_i \subseteq \biguplus_{1\leq i\leq k} TB_i = TB$, and hence, we obtain the monomorphism $\mathrm{in}_{TA} = \biguplus_{1\leq i\leq k} (\mathrm{in}_{TA_i} : TA_i \hookrightarrow TB_i)$. □

Thus, we are able to define all kinds of morphisms in **DB** category, as follows.

Proposition 25. *The following properties for morphisms are valid:*

1. *Each arrow $f : A \to B$ such that $\widetilde{f} \simeq TA$ (and in the special case when $\widetilde{f} = TA$) is a monomorphism (monic arrow).*
2. *Each arrow $f : A \to B$ such that $\widetilde{f} \simeq TB$ (and in the special case when $\widetilde{f} = TB$) is an epimorphism (epic arrow).*
3. *Each arrow $f : A \to B$ such that $\widetilde{f} \simeq TA \simeq TB$ (and in the special case when $\widetilde{f} = TA = TB$) is an isomorphism (iso arrow).*

Proof. 1. For a simple arrow, it holds from Corollary 17, so consider a strictly-complex source object A. For a given complex arrow $f : \biguplus_{1\leq j\leq m} A_j \to \biguplus_{1\leq i\leq k} B_i$, let there be an isomorphism $TA \simeq \widetilde{f}$. Then, from this isomorphism and the fact that $\widetilde{f}$ is strictly complex, also $TA = \biguplus_{1\leq j\leq m} A_j$ is strictly complex, so we obtain $m = |\frac{f}{\biguplus}|$ and a bijection $\sigma' : \frac{f}{\biguplus} \to \{1, \ldots, m\}$ such that for each $f_{ji} : A_j \to B_i, \widetilde{f}_{ji} = TA_{\sigma'(f_{ji})} = TA_j$. Thus, from above, this ptp (simple) arrow f_{ji} is monic.

For each pair of morphisms $g, h : \biguplus_{1\leq i\leq k} X_i \to A$ (with $(g_{lj} : X_l \to A_j) \in \frac{g}{\biguplus} \neq \emptyset$ and $(h_{lj} : X_l \to A_j) \in \frac{h}{\biguplus} \neq \emptyset$, it holds from above that $\widetilde{g_{lj}} \subseteq TA_j$ and $\widetilde{h_{lj}} \subseteq TA_j$) if $f \circ g = f \circ h : X \to B$ then, from point 3 of Definition 44, $\frac{f\circ g}{\biguplus} = \frac{f\circ h}{\biguplus}$, i. e., they have the same ptp arrows, i. e., $f_{ji} \circ g_{lj} = f_{ji} \circ h_{lj} : X_l \to B_i$, and hence, they have the same fluxes, i. e., $\widetilde{f_{ji} \circ g_{lj}} = \widetilde{f_{ji} \circ h_{lj}}$. However, $\widetilde{f_{ji} \circ g_{lj}} = \widetilde{f}_{ji} \bigcap \widetilde{g}_{lj} = TA_j \bigcap \widetilde{g}_{lj} = \widetilde{g}_{lj}$ and $\widetilde{f_{ji} \circ h_{lj}} = \widetilde{f}_{ji} \bigcap \widetilde{h}_{lj} = TA_j \bigcap \widetilde{h}_{lj} = \widetilde{h}_{lj}$, so that $\widetilde{h}_{lj} = \widetilde{g}_{lj}$, i. e., $h_{lj} = g_{lj} : X_l \to A_j$. It holds for all ptp arrows in g and h so that $g = h$ and, consequently, f is an monomorphism. If A is not strictly complex, then we have an isomorphism (which is monic as well) is $: A \to A'$ (thus, $TA \simeq TA'$) where A' is strictly complex, so that $f = f' \circ \mathrm{is} : A \to B$ (where $f' : A' \to B$ is a monomorphism above with $\widetilde{f'} \simeq TA' \simeq TA$) is a composition of two monomorphism, with $\widetilde{f} \simeq \widetilde{f'}$, i. e., $\widetilde{f} \simeq TA$, and hence, it is monic as well.

Claim 2. For a simple arrow, it holds from Corollary 17 and so consider a strictly complex target object B. For a given complex arrow $f : \biguplus_{1\leq j\leq m} A_j \to \biguplus_{1\leq i\leq k} B_i$, let there be an isomorphism $TB \simeq \widetilde{f}$. Then, from this isomorphism and the fact that $\widetilde{f}$ and also $TB = \biguplus_{1\leq j\leq k} B_j$ are strictly complex, we obtain $k = |\frac{f}{\biguplus}|$ and bijection $\sigma' : \frac{f}{\biguplus} \to \{1, \ldots, k\}$ such that for each $f_{ji} : A_j \to B_i, \widetilde{f}_{ji} = TB_{\sigma'(f_{ji})} = TB_i$. Thus, from above, this ptp (simple) arrow f_{ji} is epic. For each pair of morphisms $g, h : B \to \biguplus_{1\leq i\leq k} X_i$ (with $(g_{il} : B_i \to X_l) \in \frac{g}{\biguplus} \neq \emptyset$ and $(h_{il} : B_i \to X_l) \in \frac{h}{\biguplus} \neq \emptyset$, it holds from above that $\widetilde{g_{il}} \subseteq TB_i$ and $\widetilde{h_{il}} \subseteq TB_i$) if $g \circ f = h \circ f : A \to X$, then, from point 3 of Definition 44, $\frac{g\circ f}{\biguplus} = \frac{h\circ f}{\biguplus}$, i. e., they have the same ptp arrows, i. e., $g_{il} \circ f_{ji} = h_{il} \circ f_{ji} : A_j \to X_l$, and hence, they have the same fluxes, i. e., $\widetilde{g_{il} \circ f_{ji}} = \widetilde{h_{il} \circ f_{ji}}$. However, $\widetilde{g_{il} \circ f_{ji}} = \widetilde{g}_{il} \bigcap \widetilde{f}_{ji} = \widetilde{g}_{il} \bigcap TB_i = \widetilde{g}_{il}$ and $\widetilde{h_{il} \circ f_{ji}} = \widetilde{h}_{il} \bigcap \widetilde{f}_{ji} = \widetilde{h}_{il} \bigcap TB_i = \widetilde{h}_{il}$, so that $\widetilde{h}_{il} = \widetilde{g}_{il}$, i. e., are equal ptp arrows $h_{il} = g_{il} : B_i \to X_l$. It holds for all ptp arrows in g and h so that $g = h$ and, consequently, f is an epimorphism. If B is not strictly complex, then we have an isomorphism (which is epic as well) is $: B' \to B$ (thus, $TB' \simeq TB$) where B' is strictly complex, so that $f = \mathrm{is} \circ f' : A \to B$ (where $f' : B' \to B$ is

an epimorphism above with $\widetilde{f'} \simeq TB \simeq TB'$) is a composition of two epimorphism, with $\widetilde{f} \simeq \widetilde{f'}$, i. e., $\widetilde{f} \simeq TB$, and hence, it is epic as well.

Claim 3. For a simple arrow, it holds from Corollary 17.

So, let us consider the strictly complex objects $A = \biguplus_{1\leq j\leq m} A_j$ and $B = \biguplus_{1\leq i\leq k} B_i$ so that $m = k \geq 2$, and from the isomorphism $TA = \biguplus_{1\leq j\leq m} A_j \simeq TB = \biguplus_{1\leq i\leq k} B_i$, we have a bijection (permutation) $\sigma : \{1,\ldots,m\} \to \{1,\ldots,k\}$ such that $TA_j = TB_{\sigma(j)}$, for $j = 1,\ldots,m$. Let us consider the most general case when $TA_j \neq TA_i$ for all $i \neq j$, $i,j \in \{1,\ldots,m\}$ (i. e., all TA_j are distinct). From $\widetilde{f} \simeq TA$, the number of ptp arrows in f has to be equal to m, and hence, $\frac{f}{\biguplus} = \{f_1,\ldots,f_m\}$ with $\widetilde{f} = \biguplus_{1\leq i\leq m} \widetilde{f_i} \simeq TA$; thus, with a permutation $\sigma_1 : \{1,\ldots,m\} \to \{1,\ldots,k\}$ such that $\widetilde{f_i} = TA_{\sigma_1(i)} = TB_{\sigma(\sigma_1(i))}$, for $i = 1,\ldots,m$. Thus, from the fact that all TA_j are distinct values, each ptp arrow is

(a) $f_i : A_{\sigma_1(i)} \to B_{\sigma(\sigma_1(i))}$.

Let us define a morphism $g : TA \to TB$ such that $\frac{g}{\biguplus} = \{g_{j\sigma(j)} : TA_j \to TB_{\sigma(j)} \mid 1 \leq j \leq m\}$ where each ptp arrow $g_{j\sigma(j)}$ is the identity arrow id_{TA_j} (from the fact that $TA_j = TB_{\sigma(j)}$). From $\frac{g^{-1}}{\biguplus} = \{g^{-1}_{j\sigma(j)} : TB_{\sigma(j)} \to TA_j \mid 1 \leq j \leq m\}$, we obtain that $g \circ g^{-1} = \mathrm{id}_{TB}$ and $g^{-1} \circ g = \mathrm{id}_{TA}$, and hence, g is an isomorphism.

It is easy to show that $\frac{\mathrm{is}_B^{-1} \circ g \circ \mathrm{is}_A}{\biguplus} = \frac{f}{\biguplus}$ where $\mathrm{is}_A = \biguplus_{1\leq j\leq m} \mathrm{is}_{A_j} : A_j \to TA_j$ and $\mathrm{is}_B^{-1} = \biguplus_{1\leq i\leq m} \mathrm{is}^{-1}_{B_i} : TB_i \to B_i$, so that for each $1 \leq j \leq m$,

(b) $f_{\sigma_1^{-1}(j)} = \mathrm{is}^{-1}_{B_{\sigma(j)}} \circ g_{j\sigma(j)} \circ \mathrm{is}_{A_j} : A_j \to B_{\sigma(j)}$ (it is equal to (a) for $i = \sigma_1^{-1}(j)$, i. e., $j = \sigma_1(i)$).

Consequently, from point 3 of Definition 44, $f = \mathrm{is}_B^{-1} \circ g \circ \mathrm{is}_A$, i. e., f is a composition of three isomorphisms, and hence, it is an isomorphism $f : A \simeq B$ as well.

Let us show that this is valid also when A and B are not strictly complex: Let A' and B' be the strictly complex objects obtained from A and B by elimination of all $\perp^0$. Then, from point 4 of Lemma 3 and commutativity (point 2 of Lemma 3), $f_1 : A \simeq A'$ is an isomorphism with $TA \simeq TA'$ and $f_2 : B' \simeq B$ is an isomorphism with $TB \simeq TB'$. Thus, we obtain $TA' \simeq TA \simeq TB \simeq TB'$, i. e., $TA' \simeq TB'$, so that $f' : A' \to B'$ for $\widetilde{f'} \simeq TA' \simeq TB'$ is an isomorphism and $f = f_2 \circ f' \circ f_1$ is a composition of the isomorphisms, and hence, an isomorphism as well. □

Notice that in this proposition we provided the *sufficient* conditions for monic, epic and isomorphic arrows in **DB** and not the necessary conditions: we have seen other cases of monic and epic arrows in the case of complex arrows, e. g., $[\mathrm{id}_A, \mathrm{id}_A] : A \biguplus A \to A$, which *are not* isomorphic. We can have the cases when A and B are isomorphic objects also when $TA \neq TB$ (as in the case when C is isomorphic to the separation-composed object $C \biguplus \perp^0$ in Corollary 3). In what follows, we will show that any two isomorphic simple objects (simple databases) in **DB** are observationally equivalent.

Remark. For each of the two objects A and B, $A \simeq B$ iff $TA \simeq TB$.

In fact, if is : $A \to B$ is an isomorphism, then $\mathrm{is}_B \circ \mathrm{is} \circ \mathrm{is}_A^{-1} : TA \to TB$ is a composition of three isomorphisms, and hence, it is an isomorphism as well. Vice versa, if is : $TA \to TB$ is an isomorphism, then $\mathrm{is}_B^{-1} \circ \mathrm{is} \circ \mathrm{is}_A : A \to B$ is a composition of three isomorphisms, and hence, it is an isomorphism as well.

Example 23. Consider the schema $\mathcal{A} = (S_A, \Sigma_A)$ with $S_A = \{\mathtt{StudentRome}(x_1, x_2), \mathtt{StudentMilan}(x_1, x_2), \mathtt{StudentFlorence}(x_1, x_2), \mathtt{StudentRMF}(x_1, x_2)\}$ and with binary relational symbols for students in the cities Rome, Milan and Florence (that associate each student name with a student id) and for all students in these three cities, and the schema $\mathcal{B} = (S_B, \emptyset)$ such that $S_B = \{\mathtt{Student}(x_1, x_2)\}$ intended to unify all students from these three cities. Σ_A is a set of tuple-generating constraints $\{\forall x_1 \forall x_2(\mathtt{StudentRome}(x_1, x_2) \wedge \neg\mathtt{StudentRMF}(x_1, x_2) \Rightarrow r_\top(\overline{0}, \overline{1})), \forall x_1 \forall x_2(\mathtt{StudentMilan}(x_1, x_2) \wedge \neg\mathtt{StudentRMF}(x_1, x_2) \Rightarrow r_\top(\overline{0}, \overline{1})), \forall x_1 \forall x_2(\mathtt{StudentFlorence}(x_1, x_2) \wedge \neg\mathtt{StudentRMF}(x_1, x_2) \Rightarrow r_\top(\overline{0}, \overline{1}))\}$. Consequently, we will consider only mapping interpretations α such that $\alpha(\mathtt{Student}) = \alpha(\mathtt{StudentRMF}) = \alpha(\mathtt{StudentRome}) \bigcup \alpha(\mathtt{StudentMilan}) \bigcup \alpha(\mathtt{StudentFlorence})$. Then $TA = T(\alpha(\mathtt{StudentRome}) \bigcup \alpha(\mathtt{StudentMilan}) \bigcup \alpha(\mathtt{StudentFlorence}) \bigcup \alpha(\mathtt{StudentRFM})) = T(\alpha(\mathtt{Student})) = TB$ and, consequently, the database-instances A and B are isomorphic, i. e., $A \simeq B$.

Let us consider how to define the isomorphic arrow $f : A \to B$ in the category **DB**.

First of all, while in the construction of TA, we can use the *union* of conjunctive queries (i. e., the *disjunctive* logical formula $\mathtt{StudentRome}(x_1, x_2) \vee \mathtt{StudentMilan}(x_1, x_2) \vee \mathtt{StudentFlorence}(x_1, x_2)$), as explained in Section A.3.2 in the Introduction, we cannot use the schema mapping $\forall x_1 \forall x_2((\mathtt{StudentRome}(x_1, x_2) \vee \mathtt{StudentMilan}(x_1, x_2) \vee \mathtt{StudentFlorence}(x_1, x_2)) \Rightarrow \mathtt{Student}(x_1, x_2))$ because the left-hand sides of implications in tgds (and SOtgds) are only conjunctive formulae.

Consequently, we have to make a decomposition of such a disjunction into the three conjunctive components. Hence, the SOtgd for this schema mapping is a formula Φ equal to $\forall x_1 \forall x_2(\mathtt{StudentRome}(x_1, x_2) \Rightarrow \mathtt{Student}(x_1, x_2)) \wedge \forall x_1 \forall x_2(\mathtt{StudentMilan}(x_1, x_2) \Rightarrow \mathtt{Student}(x_1, x_2)) \wedge \forall x_1 \forall x_2(\mathtt{StudentFlorence}(x_1, x_2) \Rightarrow \mathtt{Student}(x_1, x_2))$, and, consequently, we define the mapping $\mathcal{M}_{AB} = \{\Phi\} : \mathcal{A} \to \mathcal{B}$.

Thus, we obtain the sketch's mapping $\mathbf{M}_{AB} = \mathrm{MakeOperads}(\mathcal{M}_{AB}) = \{q_1, q_2, q_3, 1_{r_\emptyset}\} : \mathcal{A} \to \mathcal{B}$ with $q_1 \in O(\mathtt{StudentRome}, \mathtt{Student}), q_2 \in O(\mathtt{StudentMilan}, \mathtt{Student}), q_3 \in O(\mathtt{StudentFlorence}, \mathtt{Student})$.

Consequently, we obtain the isomorphic arrow $f = \alpha^*(\{q_1, q_2, q_3, 1_{r_\emptyset}\}) : A \to B$ where $f = \{\alpha(q_1), \alpha(q_2), \alpha(q_3), q_\perp\}$ is the set of injective functions:

$$\mathrm{in}_1 = \alpha(q_1) : \alpha(\mathtt{StudentRome}) \to \alpha(\mathtt{Student}),$$
$$\mathrm{in}_2 = \alpha(q_2) : \alpha(\mathtt{StudentMilan}) \to \alpha(\mathtt{Student}),$$
$$\mathrm{in}_3 = \alpha(q_3) : \alpha(\mathtt{StudentFlorence}) \to \alpha(\mathtt{Student}),$$

with the information flux (here $\mathbf{x}$ denotes the tuple (x_1, x_2)) equal to:

$$\begin{aligned}\widetilde{f} &= B_{\top}(f)\\ &= T\{\pi_{\mathbf{x}}\|\texttt{StudentRome}\|_A, \pi_{\mathbf{x}}\|\texttt{StudentMilan}\|_A, \pi_{\mathbf{x}}\|\texttt{StudentFlorence}\|_A\}\\ &= T\{\alpha(\texttt{StudentRome}), \alpha(\texttt{StudentMilan}), \alpha(\texttt{StudentFlorence})\}\\ &= TA,\end{aligned}$$

i. e., it is monic and, from the fact that $TA = TB$, and hence, $\widetilde{f} = TB$, it is also epic. Consequently, from Proposition 25, $f : A \to B$ is an isomorphism.

If we enlarge the schema $\mathcal{B}$ with a new relation $\texttt{Takes1}(x_1, x_3)$ that associates student names with courses such that $\alpha(\texttt{Takes1})$ is not the empty relation, then $\widetilde{f} = TA \subset TB$, and hence, in this case the morphism above $f : A \to B$ is a monic arrow only.

Remark. It is important to take clearly in mind that the equality of arrows in **DB**, and hence, the commutativity of the diagrams in **DB** is based on the observational equivalence in Definition 44 and not on the set-theoretic equality. The (observational) equality of morphisms is more general then the set-theoretic equality.

Corollary 18. *For each two set-theoretically equal simple morphisms $f, g : A \to B$, they are equal in category* **DB** *(in sense of observational equivalence in Definition* 44*) but not vice versa.*

Proof. If f and g are set theoretically equal, then they have the same operads mapping $\mathbf{M}_{AB}$, and hence, they have the same information flux $\text{Flux}(\alpha, \mathbf{M}_{AB})$, i. e., $\widetilde{f}$ is set theoretically equal to $\widetilde{g}$ and, from Definition 44, $f = g$ in **DB** as well. Let us consider the SOtgd for this schema mapping $\mathcal{M}'_{AB} = \{\forall x_1 \forall x_2(\texttt{StudentRMF}(x_1, x_2) \Rightarrow \texttt{Student}(x_1, x_2))\} : \mathcal{A} \to \mathcal{B}$, in Example 23 above, with $\mathbf{M}'_{AB} = \text{MakeOperads}(\mathcal{M}'_{AB}) = \{q_4, 1_{r_\emptyset}\} : \mathcal{A} \to \mathcal{B}$ and $q_4 \in O(\texttt{StudentRMF}, \texttt{Student})$. Then we obtain the isomorphism $f_1 = \alpha^*(\{q_4, 1_{r_\emptyset}\}) : A \to B$ where $f_1 = \{\alpha(q_4), q_\perp\}$ with the injective function $\text{in}_4 = \alpha(q_1) : \alpha(\texttt{StudentRMF}) \to \alpha(\texttt{Student})$, so that $\widetilde{f_1} = \widetilde{f}$, i. e., $f_1 = f$ in **DB**, while they are not set theoretically equal. □

We obtained that only for the complex arrows we have the situation that a given morphism, which is both monic and epic is not also an isomorphism, and hence, **DB** is not a topos. So, in what follows, we will explain what exactly the particular condition is, when a complex arrow, which is monic and epic, is also an isomorphism in the **DB** category.

Notice that an empty arrow $\perp^1 = \{q_\perp : \perp \to \perp\}$ between two databases, with $\widetilde{\perp^1} = \perp^0 = \{\perp\}$, is equivalent to the absence of the schema mapping between these two databases. In fact, because of that, we are always able to introduce the empty arrows between databases, as it was used in Lemma 2 for the construction of canonical complex morphisms in Section 7.1. In fact, an empty arrow is not a ptp arrow, so that the presence or absence of them in a given complex arrow does not change such an arrow (from Definition 44 two complex arrows are equal iff they have the same set of ptp arrows and the empty arrows are on ptp arrows). Moreover, note that each empty arrow $\perp^1 : A_j \to B_i$ is

not monic if $A_j \neq \perp^0$ (to be monic it must satisfy $TA_j = \widetilde{\perp^1} = \perp^0$) and is not epic if $B_i \neq \perp^0$ (to be epic it must satisfy $TB_i = \widetilde{\perp^1} = \perp^0$), and is not an isomorphism if $A_j \neq \perp^0$ or $B_i \neq \perp^0$.

Notice also that, from the fact that $A \biguplus \perp^0$ is isomorphic to A, we can replace any original complex-object B by a *strictly complex* object B' obtained by elimination of the empty objects $\perp^0$ from the disjoint union. Thus, any complex arrow $g : A \to B$ can be substituted by an equivalent arrow $f : A' \to B'$ between two strictly complex objects A' and B' such that $\frac{f}{\biguplus} = \frac{g}{\biguplus}$ so that f can always be well-defined from the set $\frac{g}{\biguplus}$ by using the canonical representation of g between strictly complex objects A' and B' (by Lemma 2).

Consequently, any proof for the morphisms can be done by using its canonical morphism between the strictly complex objects obtained from the original source and target objects (as we did in the last part of the proof of Proposition 25).

Lemma 4. *Each complex arrow $f : A \to B$, between the strictly complex objects $A = \biguplus_{1 \leq j \leq m} A_j$, $B = \biguplus_{1 \leq i \leq k} B_i$ with $m, k \geq 2$, is an isomorphism iff it is both monic and epic and composed of $m = k$ point-to-point arrows.*

Proof. From Proposition 23 and the fact that f is monic, for each A_j must exist a ptp arrow $f_{ji} \in \frac{f}{\biguplus}$ with $\mathrm{dom}(f_{ji}) = A_j$, $|\frac{f}{\biguplus}| \geq m$. Moreover, f is epic, and hence, for each B_i must exist a ptp arrow $f_{li} \in \frac{f}{\biguplus}$ with $\mathrm{cod}(f_{li}) = B_i$, so that $|\frac{f}{\biguplus}| \geq k$. Thus, the fact that $|\frac{f}{\biguplus}| = m = k$ is consistent with the facts that f is monic and epic, and for each A_j we have only one outgoing ptp arrow from it while for each B_i we have only one ingoing arrow into it. Moreover, from the fact that f is both monic and epic, each of its ptp arrows $(f_{ji} : A_j \to B_i) \in \frac{f}{\biguplus}$ must be epic and monic (by Proposition 23), and from the fact that it is a simple arrow, it must be an isomorphism with $\widetilde{f_{ji}} = TA_j = TB_i$ (i. e., $A_j \simeq B_i$), with its inverse $f_{ji}^{\mathrm{OP}} : B_i \to A_j$. Consequently, from Definition 43, $\frac{f \circ f^{\mathrm{OP}}}{\biguplus} = \{f_{ji} \circ f_{ji}^{\mathrm{OP}} \mid f_{ji} \in \frac{f}{\biguplus}\} = \{\mathrm{id}_{B_i} \mid f_{ji} \in \frac{f}{\biguplus}\} =$ (from the fact that f is epic, for each B_i it must exist a ptp arrow f_{ji} with $\mathrm{cod}(f_{ji}) = B_i$) $= \{\mathrm{id}_{B_i} \mid 1 \leq i \leq m\} = \frac{\mathrm{id}_B}{\biguplus}$. Thus, from Definition 44, we obtain $f \circ f^{\mathrm{OP}} = \mathrm{id}_B$. Analogously, $\frac{f^{\mathrm{OP}} \circ f}{\biguplus} = \{f_{ji}^{\mathrm{OP}} \circ f_{ji} \mid f_{ji} \in \frac{f}{\biguplus}\} = \{\mathrm{id}_{A_j} \mid f_{ji} \in \frac{f}{\biguplus}\} =$ (from the fact that f is monic, for each A_j it must exist a ptp arrow f_{ji} with $\mathrm{dom}(f_{ji}) = A_j$) $= \{\mathrm{id}_{A_j} \mid 1 \leq i \leq m\} = \frac{\mathrm{id}_A}{\biguplus}$. Thus, from Definition 44, we obtain $f^{\mathrm{OP}} \circ f = \mathrm{id}_A$. Consequently, f is an isomorphism. □

Let us now generalize this result to all complex arrows.

Corollary 19. *Each complex arrow $f : A \to B$ is an isomorphism iff it is both monic and epic and the strictly complex objects obtained from A and B are composed by $n = |\frac{f}{\biguplus}|$ of nonempty simple databases.*

Proof. Let A' and B' be the strictly complex objects obtained from A and B, respectively, with the isomorphisms $\mathrm{is}_1 : A \to A'$ and $\mathrm{is}_2 : B \to B'$ (which are always monic and epic). If f is monic, then $f' = \mathrm{is}_2 \circ f \circ \mathrm{is}_1^{-1} : A' \to B'$ is monic from the fact that the composition of monic arrows is monic. Analogously, if f is epic then f' is epic. Consequently, if f is

both monic and epic, then f' is both monic and epic, so that if A' and B' are composed by the same number of nonempty objects (equal to $|\frac{f'}{\downarrow}|$), from Lemma 4 we obtain that f' is an isomorphism. Thus, from the fact that $f = \mathrm{is}_2^{-1} \circ f' \circ \mathrm{is}_1$ is a composition of the three isomorphisms, it is an isomorphism as well. This is valid also when we obtain for the strictly complex objects $A' = B' = \perp^0$ of the objects A and B, and in such a case $n = 0$. □

Let us introduce the additional properties for the arrows in **DB** as principal morphisms (used for definition of "exponent" objects, Proposition 32), retraction pairs (for PO relation $\preceq$ and subcategory $\mathbf{DB}_I$, Theorem 17) and the idempotents (for conceptual category symmetry, Theorem 14).

Proposition 26. *The following properties for arrows in* **DB** *are valid:*

1. *For any two objects A and B, the arrow $h : A \longrightarrow B$ such that $\widetilde{h} \simeq A \otimes B$ is a principal morphism.*
2. *For any monomorphism $f : A \hookrightarrow B$ and its inverted arrow (an epimorphism) $f^{\mathrm{OP}} : B \twoheadrightarrow A$, the pair (f, f^{OP}) is a retraction pair.*
3. *For any simple object A, there is a category of idempotents on A, denoted by Ret_A and defined as follows:*
   3.1. *Objects of Ret_A form the set of all simple arrows from A into A, i. e., $\mathrm{Ob}_{\mathrm{Ret}_A} = \mathbf{DB}(A, A)$.*
   3.2. *For any two objects $f, g \in \mathrm{Ob}_{\mathrm{Ret}_A}$, each arrow between them is a pair $(h; h) : f \to g$ where $h : A \to A$ with $\widetilde{h} \subseteq \widetilde{f} \bigcap \widetilde{g}$, and hence, there is the bijection $\sigma : \mathrm{Ret}_A(f, g) \simeq \mathbf{DB}(\widetilde{f}, \widetilde{g})$ such that for any $(h; h) \in \mathrm{Ret}_A(f, g)$, $\sigma(h; h) = T_e(h; h)$ for the symmetry-closure functor T_e in Theorem 14.*

Proof. Claim 1. For any simple arrow $h : A \longrightarrow B$ with $\widetilde{h} = A \otimes B$, we have $\forall f : A \longrightarrow B$, $\exists g : A \longrightarrow A$ such that $f = h \circ g$. In fact, it is satisfied for $g \approx f$, i. e., $g = \{\mathrm{id}_R : R \to R \mid R \in \widetilde{f}\} \bigcup \{q_\perp\}$. Let h be a complex arrow with a set of point-to-point arrows $h_{ji} : A_j \to B_i$ in $\frac{h}{\downarrow}$. Hence, from the fact that $\widetilde{h} = A \otimes B$, we obtain $\widetilde{h_{ji}} = A_j \otimes B_i = TA_j \bigcap TB_i$. For any arrow $f : A \to B$, from Proposition 24 we have that $\widetilde{f} \preceq A \otimes B$ and, consequently, for each of its point-to-point arrows $f_{li} : A_l \to B_i$, $\widetilde{f_{li}} \subseteq A_l \otimes B_i = TA_l \bigcap TB_i = \widetilde{h_{li}} \subseteq TA_l$. Thus, for this ptp arrow f_{li} we define a ptp arrow $g_{ll} = \{\mathrm{id}_R : R \to R \mid R \in \widetilde{f_{li}} \subseteq TA_l\} \bigcup \{q_\perp\} : A_l \to A_l$, so that $\widetilde{g_{ll}} = \widetilde{f_{li}}$ and for composition $h_{li} \circ g_l : A_l \to B_i$, $\widetilde{h_{li} \circ g_{ll}} = \widetilde{h_{li}} \bigcap \widetilde{g_{ll}} = (TA_l \bigcap TB_i) \bigcap \widetilde{g_{ll}} = (A_l \otimes B_i) \bigcap \widetilde{f_{li}} = \widetilde{f_{li}}$. By this generation of ptp arrows g_{ll} for each ptp arrow in $\frac{f}{\downarrow}$, we define the canonical morphism $g : A \to A$ (in Lemma 2), such that $\frac{f}{\downarrow} = \frac{h \circ g}{\downarrow}$, i. e., from Definition 44, $f = h \circ g : A \to B$.

Claim 2. For any simple arrow, which is a monomorphism $f : A \hookrightarrow B$ and (by duality, Theorem 12) the epimorphism $f^{\mathrm{OP}} : B \twoheadrightarrow A$ ($f^{\mathrm{OP}} \approx f$), $f^{\mathrm{OP}} \circ f = \mathrm{id}_A$ (in fact, $\widetilde{f^{\mathrm{OP}} \circ f} = \widetilde{f^{\mathrm{OP}}} \bigcap \widetilde{f} = \widetilde{f} = TA = \widetilde{\mathrm{id}_A}$).

For any complex monomorphism between two strictly-complex objects

$$f : \biguplus_{1\le j\le m} A_j \hookrightarrow \biguplus_{1\le i\le k} B_i,$$

from Proposition 23, $\frac{f}{\biguplus} = \{f_{ji_j} : A_j \hookrightarrow B_{i_j} \mid 1 \le j \le m\}$, so that $\frac{f^{\mathrm{OP}}}{\biguplus} = \{f_{ji_j}^{\mathrm{OP}} : B_{i_j} \twoheadrightarrow A_j \mid 1 \le j \le m\}$, and hence, $\frac{f^{\mathrm{OP}} \circ f}{\biguplus} = \{h_{jj} = f_{ji_j}^{\mathrm{OP}} \circ f_{ji_j} = \mathrm{id}_{A_j} : A_j \to A_j \mid 1 \le j \le m\}$, i. e., $f^{\mathrm{OP}} \circ f = \mathrm{id}_A$. If A and B are not strictly complex, let A' and B' be their strictly complex reductions with two isomorphisms $\mathrm{is}_1 : A \to A'$ and $\mathrm{is}_2 : B \to B'$ and the monomorphism $f_1 : A' \hookrightarrow B'$. Then $f = \mathrm{is}_2 \circ f_1 \circ \mathrm{is}_1$ with $f^{\mathrm{OP}} = \mathrm{is}_1^{-1} \circ f_1^{\mathrm{OP}} \circ \mathrm{is}_2^{-1}$, and hence, $f^{\mathrm{OP}} \circ f = \mathrm{is}_1^{-1} \circ f_1^{\mathrm{OP}} \circ \mathrm{is}_2^{-1} \circ \mathrm{is}_2 \circ f_1 \circ \mathrm{is}_1 = \mathrm{is}_1^{-1} \circ f_1^{\mathrm{OP}} \circ \mathrm{id}_{B'} \circ f_1 \circ \mathrm{is}_1 = \mathrm{is}_1^{-1} \circ f_1^{\mathrm{OP}} \circ f_1 \circ \mathrm{is}_1 =$ (from the strictly complex composition above) $= \mathrm{is}_1^{-1} \circ \mathrm{id}_{A'} \circ \mathrm{is}_1 = \mathrm{is}_1^{-1} \circ \mathrm{is}_1 = \mathrm{id}_A$.

Claim 3. For each $f : A \hookrightarrow A, f \circ f = f$, and hence, it is idempotent and, consequently, an object in Ret_A. The identity arrow of an object f in Ret_A is the arrow $\mathrm{id}_f = (f; f) : f \to f$. Given two arrows $(h; h) : f \to g$ and $(h_1; h_1) : g \to k$, and hence, $\widetilde{h} \subseteq \widetilde{f} \bigcap \widetilde{g}$ and $\widetilde{h_1} \subseteq \widetilde{g} \bigcap \widetilde{k}$, their composition is an arrow $(h_1; h_1) \circ (h; h) = (h_1 \circ h; h_1 \circ h) : f \to k$ because $\widetilde{h_1 \circ h} = \widetilde{h_1} \bigcap \widetilde{h} \subseteq (\widetilde{f} \bigcap \widetilde{g}) \bigcap (\widetilde{g} \bigcap \widetilde{k}) \subseteq \widetilde{f} \bigcap \widetilde{k}$.

The mapping $\sigma : \mathrm{Ret}_A(f, g) \simeq \mathbf{DB}(\widetilde{f}, \widetilde{g})$ is well-defined. In fact, for any arrow in Ret_A, $(h; h) : f \to g$, from Theorem 14, $T_e(h; h) = \mathrm{in}_{\widetilde{g}}^{\mathrm{OP}} \circ T(h \circ f) \circ \mathrm{in}_{\widetilde{g}} : \widetilde{f} \to \widetilde{g}$, so that $\widetilde{T_e(h; h)} = \widetilde{\mathrm{in}_{\widetilde{g}}^{\mathrm{OP}}} \bigcap \widetilde{T(h \circ f)} \bigcap \widetilde{\mathrm{in}_{\widetilde{g}}} = \widetilde{g} \bigcap (\widetilde{f} \bigcap \widetilde{h}) \bigcap \widetilde{f} = \widetilde{h} \subseteq \widetilde{g} \bigcap \widetilde{f}$ is appropriate. The inverse of σ is given by $\sigma^{-1}(T_e(h; h)) = (k; k)$ where $k \triangleq \tau_{J(g)}^{-1} \circ h \circ \tau_{J(f)} : A \to A$ such that $\widetilde{k} = \widetilde{\tau_{J(g)}^{-1}} \bigcap \widetilde{h} \bigcap \widetilde{\tau_{J(f)}} = \widetilde{g} \bigcap \widetilde{h} \bigcap \widetilde{f} = \widetilde{h}$, i. e., $k = h$. □

7.1.2 Power-view endofunctor and monad T

Let us extend the power-view operator T for objects into an endofunctor of **DB** category.

Theorem 11. *There exists the endofunctor $T = (T^0, T^1) : \mathbf{DB} \longrightarrow \mathbf{DB}$ such that:*

1. *For any object A, the object component T^0 is equal to the type power-view operator T, i. e., $T^0(A) \triangleq TA$.*
2. *For any morphism $f : A \longrightarrow B$, the arrow component T^1 is defined by*

$$Tf = T^1(f) \triangleq \mathrm{is}_B \circ f \circ \mathrm{is}_A^{-1} : TA \to TB$$

   with $\frac{Tf}{\biguplus} = \{Tf_i \mid f_i \in \frac{f}{\biguplus}\}$, so that $\widetilde{Tf} \simeq \widetilde{f}$ and $T(Tf) = Tf$.
3. *Functor T preserves the properties of arrows, i. e., if a morphism f has a property $P \in \{$monic, epic, isomorphic$\}$ then also $T(f)$ has the same property P.*

Proof. From the fact that all arrows are composed of atomic arrows, it is enough to consider only them. We have for composition of morphisms (derived from a composition

of schema mappings and their information fluxes in Definition 67) that for a simple arrow f, $\widetilde{Tf} = \widetilde{\mathrm{is}_B \circ f \circ \mathrm{is}_A^{-1}} = \widetilde{\mathrm{is}_B} \bigcap \tilde{f} \bigcap \widetilde{\mathrm{is}_A^{-1}} = TB \bigcap \tilde{f} \bigcap TA =$ (from Proposition 24 if f is a simple arrow, then $\tilde{f} \subseteq TA \bigcap TB$) $= \tilde{f}$. If f is a complex arrow with $A = \biguplus_{1\le j\le m} A_j$ and $B = \biguplus_{1\le i\le k} B_i$, then $\mathrm{is}_A^{-1} = \biguplus_{1\le j\le m} \mathrm{is}_{A_j}^{-1} : TA_j \to A_j$ with $\mathrm{is}_{A_j}^{-1} = \{\mathrm{id}_R : R \to R \mid R \in A_j\}$, so that $\frac{\mathrm{is}_A^{-1}}{\biguplus} = \frac{\mathrm{id}_A}{\biguplus}$ and $\mathrm{is}_B = \biguplus_{1\le i\le k} \mathrm{is}_{B_i} : B_i \to TB_i$ with $\mathrm{is}_{B_i} = \{\mathrm{id}_R : R \to R \mid R \in B_i\}$, and hence, $\frac{\mathrm{is}_B}{\biguplus} = \frac{\mathrm{id}_B}{\biguplus}$. Consequently, for each $f'_{ji} = Tf_{ji} = \mathrm{is}_{B_i} \circ f_{ji} \circ \mathrm{is}_{A_j}^{-1} \in \frac{Tf}{\biguplus}$ with $f_{ji} : A_j \to B_i \in \frac{f}{\biguplus}$, $\widetilde{f'_{ji}} = \widetilde{f_{ji}}$ so that, from Definition 43,

$$\widetilde{Tf} \simeq \widetilde{\biguplus \frac{Tf}{\biguplus}} \simeq \widetilde{\biguplus \frac{f}{\biguplus}} \simeq \tilde{f}.$$

For each identity arrow $\mathrm{id}_A : A \to A$ with $\widetilde{\mathrm{id}_A} = TA$, we obtain the identity arrow $\mathrm{id}_{TA} = T\mathrm{id}_A : TA \to TA$ with $\widetilde{\mathrm{id}_{TA}} = \widetilde{T\mathrm{id}_A} = \widetilde{\mathrm{id}_A} = TA$. For the composition of any two arrows $f : A \to B$ and $g : B \to C$, $T(g \circ f) = \mathrm{is}_C \circ (g \circ f) \circ \mathrm{is}_A^{-1} = \mathrm{is}_C \circ g \circ \mathrm{id}_B \circ f \circ \mathrm{is}_A^{-1} = \mathrm{is}_C \circ g \circ (\mathrm{is}_B^{-1} \circ \mathrm{is}_B) \circ f \circ \mathrm{is}_A^{-1} = Tg \circ Tf$. Thus, from Definition 44, $T(g \circ f) = Tg \circ Tf$. Consequently, the power-view operator T is a **DB** endofunctor.

Thus, $T(Tf) = T(\mathrm{is}_B \circ f \circ \mathrm{is}_A^{-1}) = T\mathrm{is}_B \circ Tf \circ T\mathrm{is}_A^{-1} =$ (from Example 24 below) $= \mathrm{id}_{TB} \circ Tf \circ \mathrm{id}_{TA} = Tf$.

It is easy to verify that T is a 2-endofunctor and preserves the properties of arrows: if f is an isomorphism, then Tf is a composition of three isomorphisms, and hence, an isomorphism as well. If f is monic, then $Tf = \mathrm{is}_B \circ f \circ \mathrm{is}_A^{-1}$, i. e., a composition of three monic arrows (each isomorphic arrow is monic and epic as well), and hence, is a monomorphism as well. The proof when f is epic is similar.

The functorial properties for different cases of complex arrows are presented with details in point 2 of Corollary 23 in Section 7.3.1. □

Let us show that the morphism $Tf : TA \to TB$, for a given morphisms $f = \alpha^*(\mathrm{MakeOperads}(\mathcal{M}_{AB}))$ with $\mathcal{M}_{AB} = \{\Psi\}$, $\mathbf{M}_{AB} = \mathrm{MakeOperads}(\mathcal{M}_{AB}) = \{q_1, \ldots, q_k, 1_{r_\emptyset}\}$ and $S = \{\partial_1(q_i) \mid 1 \le i \le k\} \subseteq S_B$, corresponds to the definition of arrows in **DB** category, specified by Theorem 10, i. e., there exists a SOtgd Φ of a schema mapping such that $Tf = \alpha^*(\mathrm{MakeOperads}(\{\Phi\}))$.

We will show that Φ is just the SOtgd of the schema mapping $\mathcal{M}_{AB} : \mathcal{A} \to \mathcal{B}$, both with a mapping-interpretation α, used in definition of the morphism f. From Theorem 11 above, we obtain

$$\begin{aligned}
Tf &= \mathrm{is}_B \circ f \circ \mathrm{is}_A^{-1} \\
&= \alpha^*\big(\mathrm{MakeOperads}(\{\Phi_B\})\big) \circ \alpha^*(\mathbf{M}_{AB}) \circ \alpha^*\big(\mathrm{MakeOperads}(\Phi_A)\big) \\
&= \alpha^*\big(\mathrm{MakeOperads}(\{\Phi_B\}) \circ \mathbf{M}_{AB} \circ \mathrm{MakeOperads}(\Phi_A)\big) \\
&= \alpha^*\Big(\big(\{1_r \mid r \in S_B\} \bigcup \{1_{r_\emptyset}\}\big) \circ \mathbf{M}_{AB} \circ \big(\{1_r \mid r \in S_A\} \bigcup \{1_{r_\emptyset}\}\big)\Big) \\
&= \alpha^*\Big(\big(\{1_r \mid r \in S_B\} \bigcup \{1_{r_\emptyset}\}\big) \circ \mathbf{M}_{AB}\Big) = \alpha^*\Big(\big(\{1_r \mid r \in S_B\} \bigcup \{1_{r_\emptyset}\}\big) \circ \{q_1, \ldots, q_k, 1_{r_\emptyset}\}\Big)
\end{aligned}$$

$$= \alpha^*\Big(\big(\{1_r \mid r \notin S\} \bigcup \{1_r \mid r \in S\} \bigcup \{1_{r_\emptyset}\}\big) \circ \{q_1, \ldots, q_k, 1_{r_\emptyset}\}\Big)$$
$$= \alpha^*\Big(\big(\{1_r \mid r \in S_B\} \bigcup \{1_{r_\emptyset}\}\big) \circ \{q_1, \ldots, q_k, 1_{r_\emptyset}\}\Big)$$
$$= \alpha^*\Big(\{1_r \mid r \notin S\} \bigcup \{q_1, \ldots, q_k, 1_{r_\emptyset}\}\Big)$$
$$= \alpha^*(\text{MakeOperads}(\{\Phi\})),$$

where Φ is equal to SOtgd $\Psi \bigwedge\{\exists f_{r_i} \forall \mathbf{x}_i((f_{r_i}(\mathbf{x}_i) = 1) \Rightarrow r_i(\mathbf{x}_i)) \mid r_i \notin S\}$ and f_{r_i} is the characteristic function of the relation $\alpha(r_i)$. Consequently, from the fact that the information flux of the component $\bigwedge\{\exists f_{r_i} \forall \mathbf{x}_i((f_{r_i}(\mathbf{x}_i) = 1) \Rightarrow r_i(\mathbf{x}_i)) \mid r_i \notin S\}$ is empty, the information fluxes of f and Tf are equal, i. e., $\widetilde{Tf} = \widetilde{f}$. Thus, $Tf = f$ in the observational sense of Definition 44, while in the set-theoretic sense, we have that $f \subseteq Tf$.

Example 24. Let us consider the definition of this endofunctor for simple isomorphic arrows $\text{is}_A : A \to TA$, $\text{is}_A^{-1} : TA \to A$ where, from the proof of Proposition 25,

$$\text{is}_A^{-1} = \text{is}_A = \mathbb{S}_A = \{\text{id}_R : R \to R \mid R \in A\}.$$

Let us show that $T^1(\text{is}_A^{-1}) = T^1(\text{is}_A) = \text{is}_{TA} = \text{id}_{TA} : TA \to TA$.

In fact, from the definition of the isomorphic arrow $\text{is}_{TA} : TA \to T(TA)$ and from the idempotence $TTA = TA$ we obtain $\text{is}_{TA} = \mathbb{S}_{TA} = \{\text{id}_R : R \to R \mid R \in TA\} : TA \to TA$, i. e., $\text{is}_{TA} = \text{id}_{TA} : TA \to TA$. Analogously, $\text{is}_{TA}^{-1} = \mathbb{S}_{TA} = \{\text{id}_R : R \to R \mid R \in TA\} : T(TA) \to TA$, i. e.,

(a) $\text{is}_{TA}^{-1} = \text{is}_{TA} = \text{id}_{TA} : TA \to TA$.
From Theorem 11 above (here $B = TA$), we have:
(b) $T^1(\text{is}_A) = \text{is}_B \circ \text{is}_A \circ \text{is}_A^{-1} = (\text{from } \text{is}_A \circ \text{is}_A^{-1} = \text{id}_{TA}) = \text{is}_B \circ \text{id}_{TA} = \text{is}_{TA} \circ \text{id}_{TA} = \text{is}_{TA}$, and (here we replace A by TA nd B by A):
(c) $T^1(\text{is}_A^{-1}) = \text{is}_A \circ \text{is}_A^{-1} \circ \text{is}_{TA}^{-1} = (\text{from } \text{is}_A \circ \text{is}_A^{-1} = \text{id}_{TA}) = \text{id}_{TA} \circ \text{is}_{TA}^{-1} = \text{is}_{TA}^{-1}$. Thus, from (a), (b) and (c) we obtain:
(d) $T^1(\text{is}_A^{-1}) = T^1(\text{is}_A) = \text{is}_{TA}^{-1} = \text{is}_{TA} = \text{id}_{TA} : TA \to TA$, where $\text{id}_{TA} = T^1(\text{id}_A)$.

The endofunctor T is a right and left adjoint to identity functor I_{DB}, i. e., $T \simeq I_{\text{DB}}$. Thus, we have the equivalence adjunction $\langle T, I_{\text{DB}}, \eta^C, \eta\rangle$ where the unit $\eta^C : T \simeq I_{\text{DB}}$ (such that for any object A the arrow $\eta_A^C \triangleq \eta^C(A) = \text{is}_A^{-1} : TA \longrightarrow A$) and the counit $\eta : I_{\text{DB}} \simeq T$ (such that for any A the arrow $\eta_A \triangleq \eta(A) = \text{is}_A : A \longrightarrow TA$) are isomorphic arrows in **DB** (by duality Theorem 12 it holds that $\eta^C = \eta^{\text{OP}}$).

We have already explained that the views of a database may be seen as its *observable computations*: what we need, in order to obtain an expressive power of computations in the category **DB**, are the categorial computational properties which are, as it is well known [94], based on monads.

Proposition 27. *Power-view endofunctor* $T = (T^0, T^1) : \mathbf{DB} \longrightarrow \mathbf{DB}$ *defines the monad* (T, η, μ) *and the comonad* (T, η^C, μ^C) *in* **DB** *such that* $\eta : I_{\text{DB}} \simeq T$ *and* $\eta^C : T \simeq I_{\text{DB}}$ *are two*

natural isomorphisms, while $\mu : TT \xrightarrow{\cdot} T$ *and* $\mu^C : T \xrightarrow{\cdot} TT$ *are equal to the natural identity transformation* $\mathrm{id}_T : T \xrightarrow{\cdot} T$ *(from the idempotence,* $T = TT$*).*

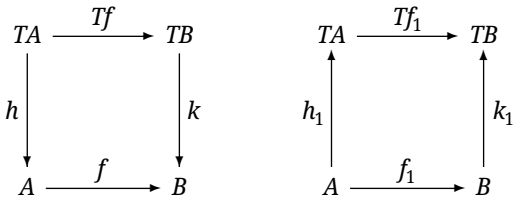
Proof. It is easy to verify that all commutative diagrams of the monad ($\mu_A \circ \mu_{TA} = \mu_A \circ T\mu_A$, $\mu_A \circ \eta_{TA} = \mathrm{id}_{TA} = \mu_A \circ T\eta_A$) and of the comonad are diagrams composed of the identity arrows. Notice that by duality (in Section 7.1.3) $\eta_{TA} = T\eta_A = \mu_A^{\mathrm{OP}}$. □

The notion of a monad is one of the most general mathematical notions. For instance, *every* algebraic theory, i. e., every set of operations satisfying equational laws, can be seen as a monad (which is also a *monoid* in a category of endofunctors of a given category: the "operation" μ being the associative multiplication of this monoid and η its unit). Thus, monoid laws of the monad do subsume all possible algebraic laws.

We use monads [12, 26, 93] to provide *denotational semantics to database mappings*, and more specifically as a way of modeling computational/collection types [94–97]. In order to interpret a database mappings (morphisms) in the category **DB**, we distinguish the object A (a database instance of type $\mathcal{A}$) from the object TA of observations (computations of type $\mathcal{A}$ without side-effects) and take as a denotation of (view) mappings the elements of TA (which are the views of type $\mathcal{A}$).

In particular, we identify A with the object of values (of type $\mathcal{A}$) and we obtain the object of observations by applying the unary type-constructor T (power-view operator) to the object A.

It is well known that each endofunctor defines the algebras and coalgebras, represented by the left and right commutative diagrams:



We will use the following well-known definitions in the category theory (we recall that the set of all arrows in a category **C** from A to B is denoted by $\mathbf{C}(A, B)$).

Definition 45. The categories CT_{alg} of T-algebras and CT_{coalg} of T-coalgebras, derived from the endofunctor T, are defined [11] as follows:

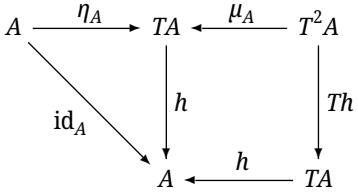
1. The objects of CT_{alg} are the pairs (A,h) with $A \in \mathrm{Ob}_{\mathrm{DB}}$ and $h \in \mathbf{DB}(TA, A)$; the arrows between objects (A,h) and (B,k) are all arrows $f \in \mathbf{DB}(A, B)$ such that $k \circ Tf = f \circ h : TA \longrightarrow B$.
2. The objects of CT_{coalg} are the pairs (A, h_1) with $A \in \mathrm{Ob}_{\mathrm{DB}}$ and $h_1 \in \mathbf{DB}(A, TA)$; the arrows between objects (A, h_1) and (B, k_1) are all arrows $f_1 \in \mathbf{DB}(A, B)$ such that $Tf_1 \circ h_1 = k_1 \circ f_1 : A \longrightarrow TB$.

The (co)algebras of an endofunctor such that it is a monad as well satisfy more structural properties and are denominated as *monadic* (co)algebras (for (co)monads introduced in Section A.2).

Definition 46. The *monadic* algebras/coalgebras are defined [11, 12] as follows:

- Each T-algebra $(A, h : TA \longrightarrow A)$ derived from a monad (T, η, μ), where h is a "structure map" such that $h \circ \mu_A = h \circ Th$ and $h \circ \eta_A = \mathrm{id}_A$, is a monadic T-algebra. The Eilenberg–Moore category of all monadic algebras T_{alg} is a full subcategory of CT_{alg}.
- Each T-coalgebra $(A, k : A \longrightarrow TA)$ derived from a comonad (T, η^C, μ^C) such that $Tk \circ k = \mu^C_A \circ k$ and $\eta^C_A \circ k = \mathrm{id}_A$ is a monadic T-coalgebra. The category of all monadic coalgebras T_{coalg} is a full subcategory of CT_{coalg}.

Thus, the monadic algebras satisfy the following commutative diagram:



In the **DB** category $\widetilde{\eta_A} = \widetilde{\mu_A} = \widetilde{\mathrm{id}_A} = TA$, while for a simple arrow h, $\widetilde{Th} = \widetilde{h} \subseteq TA \otimes A = TA$ (from Proposition 24), so that the intersection of the fluxes of the arrows in this diagram commute.

Corollary 20. *For each object A in* **DB***, there is exactly one T-algebra (and its dual T-coalgebra) $h = TA \to A$, which is equal to the isomorphism* is_A^{-1}.

Proof. If $A = \perp^0$, then $TA = \perp^0$, so that all arrows in the diagram above are the identity arrows $\mathrm{id}_{\perp^0} = \perp^1$ (which is isomorphism as well) with $h = \perp^1$.

Let $A = \biguplus_{1 \le j \le m} A_j$. Thus, $TA = \biguplus_{1 \le j \le m} TA_j$ with

$$\mu_A = T\mathrm{id}_A = \mathrm{id}_{TA} = \biguplus_{1 \le j \le m} \mathrm{id}_{TA_j} : TA_j \to TA_j$$

and the isomorphism

$$\eta_A = \biguplus_{1 \le j \le m} \eta_{A_j} = \biguplus_{1 \le j \le m} \mathrm{is}_{A_j} : A_j \to TA_j.$$

From the commutativity of the triangle above, $\mathrm{id}_A = h \circ \eta_A$, it must hold, from Definition 44, that $\frac{h \circ \eta_A}{\biguplus} = \frac{\mathrm{id}_A}{\biguplus}$, and hence, for each ptp arrow $\mathrm{id}_{A_j} : A_j \to A_J$ in $\frac{\mathrm{id}_A}{\biguplus}$, we must have the ptp arrow $h_{jj} : TA_j \to A_J$ in $\frac{h}{\biguplus}$. Consequently, $\mathrm{id}_{A_j} = h_{jj} \circ \eta_{A_j}$, i. e., $\widetilde{\mathrm{id}_{A_j}} = TA_j = \widetilde{h_{jj} \circ \eta_{A_j}} = \widetilde{h_{jj}} \bigcap \widetilde{\eta_{A_j}} = \widetilde{h_{jj}} \bigcap TA_j = \widetilde{h_{jj}}$ (from the fact that, from Proposition 24, $\widetilde{h_{jj}} \subseteq T(TA_j) \bigcap TA_j = TA_j$). Thus, from Proposition 25, $h_{jj} : TA_j \to A_J$ is an isomorphism and

$$\frac{h}{\biguplus} \supseteq \{h_{jj} : TA_j \to A_j \mid 1 \le j \le m\}.$$

However, from the fact that the number if ptp arrows in h cannot be greater that the number of ptp arrows in id_A (otherwise, the composed arrow $h \circ \eta_A = h \circ \mathrm{is}_A$ would have more ptp arrows than is_A, i. e., more ptp arrows than in id_A, and we would have that $\frac{\mathrm{id}_A}{\biguplus} \neq \frac{h \circ \eta_A}{\biguplus}$, i. e., $\mathrm{id}_A \neq h \circ \eta_A$), we obtain this unique monadic T-algebra $h = \mathrm{is}_A^{-1} : TA \to A$. □

7.1.3 Duality

The following duality theorem shows that for any commutative diagram in **DB** there is an equal commutative diagram, composed of equal objects and inverted equivalent arrows. This duality property of **DB** is a consequence of the fact that a composition of simple arrows (i. e., morphisms) is semantically based on the set-intersection commutativity of information fluxes of these arrows, and the information flux of inverted arrow is equal to the information flux of the original arrow.

Thus, *any limit diagram* in **DB** has also its equivalent *inverted colimit diagram* with equal objects and, more generally, *any universal property* also has its *equivalent couniversal property* in **DB**. From the fact that all arrows are composed of simple atomic arrows, it is enough to consider only the atomic arrows. Duality properties for different cases of complex arrows are presented with details also in point 3 of Corollary 23 in Section 7.3.1.

Theorem 12. *There exists the controvariant functor* $\underline{S} = (\underline{S}^0, \underline{S}^1) : \mathbf{DB} \longrightarrow \mathbf{DB}$ *such that:*

1. $\underline{S}^0$ *is an identity function on objects.*
2. *For any simple arrow* $f : A \longrightarrow B$ *with* $(Tf)^{\mathrm{OP}} \triangleq \{\mathrm{id}_R \mid R \in \widetilde{f}\} : TB \to TA$, *we define* $\underline{S}^1(f) \triangleq f^{\mathrm{OP}} : B \to A$, *where* $f^{\mathrm{OP}} = \mathrm{is}_A^{-1} \circ (Tf)^{\mathrm{OP}} \circ \mathrm{is}_B$ *is an inverted morphism equivalent to* f *(i. e., with* $\widetilde{f^{\mathrm{OP}}} = \widetilde{f}$*).*
   The inverse of an inverted morphism f^{OP} *is equal to the original morphism, i. e.,* $(f^{\mathrm{OP}})^{\mathrm{OP}} = f$. *For a complex arrow, we have* $\frac{f^{\mathrm{OP}}}{\biguplus} = \{f_{ji}^{\mathrm{OP}} \mid f_{ji} \in \frac{f}{\biguplus}\}$.
3. *The category* **DB** *is isomorphic to its dual category* $\mathbf{DB}^{\mathrm{OP}}$.

Proof. $\widetilde{(Tf)^{\mathrm{OP}}} = \widetilde{f}$ and $\widetilde{f^{\mathrm{OP}}} = \mathrm{is}_A^{-1} \circ \widetilde{(Tf)^{\mathrm{OP}}} \circ \mathrm{is}_B = \widetilde{\mathrm{is}_A^{-1}} \bigcap \widetilde{(Tf)^{\mathrm{OP}}} \bigcap \widetilde{\mathrm{is}_B} = TA \bigcap \widetilde{(Tf)^{\mathrm{OP}}} \bigcap TB = TA \bigcap \widetilde{f} \bigcap TB = \widetilde{f}$.

The inverted arrow of any identity arrow is equal to identity arrow. For the composition of f with $g : B \to C$, we have $S^1(g \circ f) = \mathrm{is}_A^{-1} \circ (T(g \circ f))^{\mathrm{OP}} \circ \mathrm{is}_C = \mathrm{is}_A^{-1} \circ (Tg \circ Tf)^{\mathrm{OP}} \circ \mathrm{is}_C = \mathrm{is}_A^{-1} \circ (Tf)^{\mathrm{OP}} \circ (Tg)^{\mathrm{OP}} \circ \mathrm{is}_C = \mathrm{is}_A^{-1} \circ (Tf)^{\mathrm{OP}} \circ \mathrm{is}_B \circ \mathrm{is}_B^{-1} \circ (Tg)^{\mathrm{OP}} \circ \mathrm{is}_C = S^1(f) \circ S^1(g)$. Thus, this contravariant functor is well-defined.

It is convenient to represent this contravariant functor as a covariant functor $S : \mathbf{DB}^{\mathrm{OP}} \longrightarrow \mathbf{DB}$ or a covariant functor $S^{\mathrm{OP}} : \mathbf{DB} \longrightarrow \mathbf{DB}^{\mathrm{OP}}$. It is easy to verify that for

compositions of these covariant functors $SS^{\mathrm{OP}} = I_{\mathrm{DB}}$ and $S^{\mathrm{OP}}S = I_{DB^{\mathrm{OP}}}$ w. r. t. the adjunction $\langle S, S^{\mathrm{OP}}, \phi\rangle : \mathbf{DB}^{\mathrm{OP}} \longrightarrow \mathbf{DB}$, where ϕ is a bijection: for each pair of objects A, B in **DB**, we have the bijection of hom-sets, $\phi_{A,B} : \mathbf{DB}(A, S(B)) \simeq \mathbf{DB}^{\mathrm{OP}}(S^{\mathrm{OP}}(A), B)$, i. e., $\phi_{A,B} : \mathbf{DB}(A, B) \simeq \mathbf{DB}(B, A)$, such that for any arrow $f \in \mathbf{DB}(A, B)$, $\phi_{A,B}(f) = S^1(f) = f^{\mathrm{OP}}$.

The unit and counit of this adjunction are the identity natural transformations, $\eta_{\mathrm{OP}} : I_{\mathrm{DB}} \longrightarrow SS^{\mathrm{OP}}$ and $\epsilon_{\mathrm{OP}} : S^{\mathrm{OP}}S \longrightarrow I_{DB^{\mathrm{OP}}}$, respectively, such that for any object A they return by its identity arrow. Thus, from this adjunction, we obtain that **DB** is isomorphic to $\mathbf{DB}^{\mathrm{OP}}$. □

Let us show that the morphism $(Tf)^{\mathrm{OP}} : TB \to TA$, for a given morphism $f = \alpha^*(\mathbf{M}_{AB}) : A \to B$ with $\mathbf{M}_{AB} = \mathrm{MakeOperads}(\mathcal{M}_{AB}) : \mathcal{A} \to \mathcal{B}$, is well-defined due to Theorem 10, i. e., there exists an SOtgd Φ of a schema mapping such that $(Tf)^{\mathrm{OP}} = \alpha^*(\mathrm{MakeOperads}(\{\Phi\}))$.

Let us define the set of relational symbols $S_{TB} = \{r_i \mid r_i \in \mathbb{R} \text{ and } \alpha(r_i) \in \Delta(\alpha, \mathbf{M}_{AB}) \subseteq \widetilde{f}\}$ and the α-intersection schemas $\mathcal{A}\bigcap\limits_{\alpha}\mathcal{A}$ and $\mathcal{B}\bigcap\limits_{\alpha}\mathcal{B}$ with $TA = \alpha^*(\mathcal{A}\bigcap\limits_{\alpha}\mathcal{A})$ and $TB = \alpha^*(\mathcal{B}\bigcap\limits_{\alpha}\mathcal{B})$ (in Definition 70). Then we can define the SOtgd Φ by a conjunctive formula $\bigwedge\{\forall \mathbf{x}_i(r_i(\mathbf{x}_i) \Rightarrow r_i(\mathbf{x}_i)) \mid r_i \in S_{TB}\}$.

In fact, $\mathrm{MakeOperads}(\{\Phi\}) = \{1_{r_i} \in O(r_i, r_i) \mid r_i \in S_{TB}\} \bigcup \{1_{r_\emptyset}\}$ is a sketch's mappings between schemas $\widehat{\mathcal{B}}$ and $\widehat{\mathcal{A}}$. Note that $r_\emptyset \in S_{TB}$ because $\bot \in \widetilde{f}$. Consequently,

$$\begin{aligned}(Tf)^{\mathrm{OP}} &= \alpha^*(\mathrm{MakeOperads}(\{\Phi\})) = \alpha^*\Big(\{1_{r_i} \in O(r_i, r_i) \mid r_i \in S_{TB}\} \bigcup \{1_{r_\emptyset}\}\Big)\\ &= \Big(\{\mathrm{id}_R \mid R \in \Delta(\alpha, \mathbf{M}_{AB}) \subseteq \widetilde{f}\} \bigcup \{q_\bot\}\Big) : TB \to TA.\end{aligned}$$

Thus, both inverted arrows $(Tf)^{\mathrm{OP}}$ and $f^{\mathrm{OP}} = \mathrm{is}_A^{-1} \circ (Tf)^{\mathrm{OP}} \circ \mathrm{is}_B$ (the demonstration is equal to the demonstration for arrow Tf after the proof of Theorem 11) are morphisms in the **DB** category as specified in Theorem 10.

Example 25. Let us consider the inverted arrows of the isomorphic arrows $\mathrm{is}_A : A \to TA$ and $\mathrm{is}_A^{-1} : TA \to A$.

For is_A (here $B = TA$), we have from Theorem 12 the following:

(1)
$$\begin{aligned}\mathrm{is}_A^{\mathrm{OP}} &= \mathrm{is}_A^{-1} \circ (T\mathrm{is}_A)^{\mathrm{OP}} \circ \mathrm{is}_{TA} = \text{(from (d) in Example 24)}\\ &= \mathrm{is}_A^{-1} \circ (\mathrm{id}_{TA})^{\mathrm{OP}} \circ \mathrm{id}_{TA} = \mathrm{is}_A^{-1} \circ (\mathrm{id}_{TA})^{\mathrm{OP}}\\ &= \text{(inverted arrow of an identity arrow is the same identity arrow)}\\ &= \mathrm{is}_A^{-1} \circ \mathrm{id}_{TA} = \mathrm{is}_A^{-1}.\end{aligned}$$

Analogously (here B is replaced by A and A by TA),

(2)
$$\begin{aligned}(\mathrm{is}_A^{-1})^{\mathrm{OP}} &= \mathrm{is}_{TA}^{-1} \circ \big(T\mathrm{is}_A^{-1}\big)^{\mathrm{OP}} \circ \mathrm{is}_A = \big(\text{from (d) in Example 24}\big)\\ &= \mathrm{id}_{TA} \circ (\mathrm{id}_{TA})^{\mathrm{OP}} \circ \mathrm{is}_A = (\mathrm{id}_{TA})^{\mathrm{OP}} \circ \mathrm{is}_A\\ &= \text{(inverted arrow of an identity arrow is the same identity arrow)}\\ &= \mathrm{id}_{TA} \circ \mathrm{is}_A = \mathrm{is}_A = \big(\mathrm{is}_A^{-1}\big)^{-1}.\end{aligned}$$

Thus, from (1) and (2), the label '$^{\mathrm{OP}}$' for these isomorphic arrows is equivalent to the inverse-arrow label '$^{-1}$'.

Let us show the validity of definition of the functor $\underline{S}$ and $T^1(f^{\mathrm{OP}}) = (Tf)^{\mathrm{OP}}$, based on Theorem 11, for $f : A \to B$:

(3)
$$\begin{aligned}(Tf)^{\mathrm{OP}} &= \left(\mathrm{is}_B \circ f \circ \mathrm{is}_A^{-1}\right)^{\mathrm{OP}} = \left(\mathrm{is}_A^{-1}\right)^{\mathrm{OP}} \circ f^{\mathrm{OP}} \circ \mathrm{is}_B^{\mathrm{OP}} = (\text{from (1) and (2)}) \\ &= \mathrm{is}_A \circ f^{\mathrm{OP}} \circ \mathrm{is}_B^{-1} : TB \to TA.\end{aligned}$$

Thus, by composing on the left with is_A^{-1}, we obtain $\mathrm{is}_A^{-1} \circ (Tf)^{\mathrm{OP}} = \mathrm{is}_A^{-1} \circ \mathrm{is}_A \circ f^{\mathrm{OP}} \circ \mathrm{is}_B^{-1} = \mathrm{id}_A \circ f^{\mathrm{OP}} \circ \mathrm{is}_B^{-1} = f^{\mathrm{OP}} \circ \mathrm{is}_B^{-1}$. By composing on the right with is_B, we obtain $\mathrm{is}_A^{-1} \circ (Tf)^{\mathrm{OP}} \circ \mathrm{is}_B = f^{\mathrm{OP}} \circ \mathrm{is}_B^{-1} \circ \mathrm{is}_B = f^{\mathrm{OP}}$, i. e., the definition $\underline{S}^1(f) \triangleq f^{\mathrm{OP}} : B \to A$ in Theorem 12 is valid.

From Theorem 11 (because of $f^{\mathrm{OP}} : B \to A$, A and B are exchanged),

$$T^1\left(f^{\mathrm{OP}}\right) = \mathrm{is}_A \circ f^{\mathrm{OP}} \circ \mathrm{is}_B^{-1} = (\text{from (3)}) = (Tf)^{\mathrm{OP}} : TB \to TA.$$

Remark. The object $\perp^0 = \{\perp\}$ (i. e., empty database instance) is the *zero object* (both terminal and initial) in **DB**. From any real object A in **DB**, there is a unique arrow (which is an epimorphism) $f = \alpha^*(\mathbf{M}_{AA_\emptyset}) = \alpha^*(\{1_{r_\emptyset}\}) = \{q_\perp\} = !^1 : A \twoheadrightarrow \perp^0$ and its dual arrow (which is a monomorphism) $f^{\mathrm{OP}} = !^1 : \perp^0 \hookrightarrow A$, both with empty information flux. It is easy to verify that each *empty database* (with all empty relations) is isomorphic to the zero object $\perp^0$.

Lemma 5. *The bottom object $\perp^0 = \{\perp\}$ is zero (terminal and initial) closed object in* **DB**.

Each database $A = \{R_1, \ldots, R_n, \perp\}$ with all empty relations $R_i = \{\langle\rangle\}$, $i = 1, \ldots, n$, is isomorphic to the zero object $\perp^0$. That is, $A \simeq \perp^0$.

Proof. For each simple object A and $f : A \to \perp^0$, we have from Proposition 24 that $\widetilde{f} \subseteq TA \bigcap T\perp^0 = TA \bigcap \perp^0 = \perp^0$ (From point 1 of Theorem 10), thus $f = !^1$ is a unique arrow from A into $\perp^0$, so that $\perp^0$ is a terminal object (analogously, $!^1 : \perp^0 \hookrightarrow A$ is the unique arrow, so that $\perp^0$ is an initial object as well).

Thus, for a complex object $A \biguplus B$, the unique arrow into zero object is $[!^1, !^1] : A \biguplus B \to \perp^0$ while its opposite is $\langle !^1, !^1 \rangle : \perp^0 \to A \biguplus B$.

For each database $A = \{R_1, \ldots, R_n, \perp\}$ where every $R_i = \{\langle\rangle\}$, $k = \mathrm{ar}(R_i) \geq 1$, is a k-ary empty relation, for $i = 1, \ldots, n$, we have the unique arrow $!^1 : A \to \perp^0$ (into terminal object) with $!^1 = \{q_\perp : \perp \to \perp\}$ and its unique inverted arrow $(!^1)^{\mathrm{OP}} : \perp^0 \to A$ (from initial object) with $(!^1)^{\mathrm{OP}} = \{q_\perp : \perp \to \perp\}$, hence $(!^1)^{\mathrm{OP}} \circ !^1 = !^1 : A \to A$. The identity arrow for A is $\mathrm{id}_A = \{\mathrm{id}_{R_1} : R_i \to R_i \mid R_i \in A\} \bigcup \{q_\perp\}$ with $\widetilde{\mathrm{id}_A} = !^0 = \widetilde{!^1}$ so that, from Definition 44, $(!^1)^{\mathrm{OP}} \circ !^1 = \mathrm{id}_A$ and, analogously, $!^1 \circ (!^1)^{\mathrm{OP}} = \mathrm{id}_{\perp^0}$. Thus, $!^1 : A \to \perp^0$ is an isomorphism, i. e., $A \simeq \perp^0$. □

7.2 Objects of DB: Basic operations and equivalence relations

In this section, we will introduce the basic operations for instance-databases used during the definition of database mappings between two (or more) databases. They are corresponding operators in this base **DB** category for two schema basic operations, separation † and connection $\bigoplus$, used in order to define the schema data mappings (Section A.3.5, after Definition 68). Then we also introduce a number of different equivalence relations for instance-databases:

- *Identity* relation – two instance-databases (sets of relations) A and B are identical when the set identity $A = B$ holds.
- *Isomorphism* relation '$\simeq$' in **DB**.
- *Behavioral equivalence* relation '$\approx$' in Definition 40—two instance-databases A and B are behaviorally equivalent when each view obtained from a database A can also be obtained from a database B and vice versa.
- *Weak observational equivalence* relation '$\approx_w$'—two instance-databases A and B are weakly equivalent when each "certain" view (without Skolem constants) obtained from a database A can be also obtained from a database B, and vice versa.

We will show the following inclusion relationships between them:

$$\text{`}=\text{'} \subseteq \text{`}\simeq\text{'} \subseteq \text{`}\approx\text{'} \subseteq \text{`}\approx_w\text{'}.$$

It is also possible to define other kinds of equivalences for databases. In the rest of this chapter, we will consider only the last two equivalences defined above. Consequently, from the fact that the schema separations and schema connections will be present in the sketch's categories obtained from the schema mapping graphs, we need, in order to satisfy the functorial semantics for database mapping, their corresponding operations at the instance-level (i. e., **DB** category) as well.

As we will see, these operations correspond to important database integration concepts used in practice.

7.2.1 Data federation operator in DB

The basic database operation is the DBMS's *data federation* of two database instances A and B. A federated database system is a type of metadatabase management system (DBMS), which transparently integrates multiple autonomous database systems into a *single federated database*. The constituent databases are interconnected via a computer network, and may be geographically decentralized. Since the constituent database systems remain autonomous, a federated database system is a contrastable alternative to the (sometimes daunting) task of merging together several disparate databases. A federated database, or virtual database, is the fully-integrated, logical composite of all con-

stituent databases in a federated database system. In this way, we are able to compute the queries with the relations of *both* databases. In fact, data federation technology is just used for such an integration of two previously separated databases.

Consequently, given any two databases (objects in **DB**) A and B, the federation of them (under the common DBMS) corresponds to *union*, denoted by $\overset{\bigcup}{rn}$ (i. e., the union $\bigcup$ with the renaming (*rn*) of the relational symbols in $\mathcal{A}$ and $\mathcal{B}$ that have originally the same names) of them under the same DBMS, and thus, equals to database $A \overset{\bigcup}{rn} B$.

That is, for any two schemas $\mathcal{A}$ and $\mathcal{B}$ and an interpretation α, such that $A = \alpha^*(\mathcal{A})$ and $B = \alpha^*(\mathcal{B})$, we have that $A \overset{\bigcup}{rn} B = \alpha^*(\mathcal{A} \bigoplus \mathcal{B})$, and we have the following.

Proposition 28. *For any two simple databases A and B, their federation $A \overset{\bigcup}{rn} B$ is isomorphic to their simple union $A \bigcup B$.*

Proof. Notice that for any relation in $A \bigcup B$ we have possibly a set of copies of the same relation, but only with different name: thus, $T(A \overset{\bigcup}{rn} B) = T(A \bigcup B)$, and hence, we have the isomorphism $\mathrm{is}^{-1} \circ \mathrm{is}_{rn} : A \overset{\bigcup}{rn} B \to A \bigcup B$ obtained from the isomorphisms $\mathrm{is}_{rn} : A \overset{\bigcup}{rn} B \to T(A \overset{\bigcup}{rn} B)$ and $\mathrm{is} : A \bigcup B \to T(A \bigcup B)$ (from Lemma 3). □

Thus, the data federation of two simple databases is isomorphic to the *merging* of these two databases. The merging of databases is formally defined in Section 7.4.2.

7.2.2 Data separation operator in DB

We define the separation composition of schemas $\mathcal{A}\dagger\mathcal{B} = (S_{\mathcal{A}\dagger\mathcal{B}}, \Sigma_{\mathcal{A}\dagger\mathcal{B}})$ as a *disjunctive union* with elements indexed by schema labels ($\ddot{\mathcal{A}}$ and $\ddot{\mathcal{B}}$, respectively), i. e., with $S_{\mathcal{A}\dagger\mathcal{B}} = \{(\ddot{\mathcal{A}}, r_i) \mid r_i \in S_A\} \bigcup \{(\ddot{\mathcal{B}}, r_j) \mid r_j \in S_B\}$, $\Sigma_{\mathcal{A}\dagger\mathcal{B}} = \{(\ddot{\mathcal{A}}, \phi_i) \mid \phi_i \in \Sigma_A\} \bigcup \{(\ddot{\mathcal{B}}, \psi_j) \mid \psi_j \in \Sigma_B\}$ (if $\mathcal{B} = \mathcal{A}$, then $\ddot{\mathcal{A}} = 1$ and $\ddot{\mathcal{B}} = 2$).

In what follows, we will express a finite separation composition by using the binary operator $\dagger$ and the parenthesis (e. g., $(\mathcal{A}\dagger(\mathcal{B}\dagger\mathcal{C}))\dagger\mathcal{D}$), so that in **DB** we will use such a binary version $\ddagger$ of this operator at the instance-database level.

Definition 47. The separation composition of any two instance-databases $A = \alpha^*(\mathcal{A})$ and $B = \alpha^*(\mathcal{B})$, obtained from the schemas $\mathcal{A}$ and $\mathcal{B}$, is denoted by $A \ddagger B$:

$$\begin{aligned} A \ddagger B \triangleq \alpha^*(S_{\mathcal{A}\dagger\mathcal{B}}) &= \{(\ddot{\mathcal{A}}, \alpha(r_i)) \mid r_i \in S_A\} \bigcup \{(\ddot{\mathcal{B}}, \alpha(r_j)) \mid r_j \in S_B\} \\ &= \{(\ddot{\mathcal{A}}, R_i) \mid R_i \in A\} \bigcup \{(\ddot{\mathcal{B}}, R_j) \mid R_j \in B\}. \end{aligned}$$

It corresponds to the instance-database of two separated schemas, i. e., to two mutually isolated instance-databases with separated database management systems, so that it is impossible to compute the queries with the relations from both schemas.

For any two morphisms $f : A \to C$ and $g : B \to D$, we define the morphism $(f \biguplus g) : A \ddagger B \to C \ddagger D$ as for coproducts.

The separation property for morphisms in **DB** is represented by the facts that

$$\partial_0\Big(f \Big(\!\!+\!\!\Big) g\Big) \triangleq \partial_0(f) \Big(\!\!+\!\!\Big) \partial_0(g) \quad , \partial_1\Big(f \Big(\!\!+\!\!\Big) g\Big) \triangleq \partial_1(f) \Big(\!\!+\!\!\Big) \partial_1(g).$$

Remark. For any database A, the *replication* of this database (over different DB servers) can be denoted by the separation-composed object $A‡A$ in this category **DB**. From the definition above, for any two databases A and B, $A‡B = B‡A$, i. e., as we intended, the *separation composition* for the databases *is a commutative* operation. We are able, analogously to Definition 69, to define any finite separation composition

$$‡^n(A_1, \dots, A_n) = A_1‡(A_2‡(\dots ‡A_n)\dots) \triangleq \alpha^*(\mathcal{A}_1† \cdots †\mathcal{A}_n). \qquad \square$$

From this definition, we have for any two instance databases A and B, $A‡B \neq A \uplus B$ (if $\mathcal{A} \neq \mathcal{B}$) because the disjoint union introduced above is strictly noncommutative, and in the **Set** category the objects $A \uplus B$ and $B \uplus A$ are not equal but only isomorphic. However, it can be shown that they are isomorphic objects in **DB** so that the definition of **DB** category in Theorem 10 is still valid after this introduction of the separation composition.

Lemma 6. *For any two simple instance-databases A and B,*

$$A‡B = B‡A \quad \textit{and} \quad A‡B \simeq A \Big(\!\!+\!\!\Big) B.$$

Proof. If $A = B$, the claim is clear because all relationships above are identities. From Definition 47:
(1) $A‡B = \{(\ddot{\mathcal{A}}, \phi_i) \mid \phi_i \in \Sigma_A\} \bigcup \{(\ddot{\mathcal{B}}, \psi_j) \mid \psi_j \in \Sigma_B\} = B‡A.$

Let us show that $\mathrm{id}_A \uplus \mathrm{id}_B$ (where $\mathrm{id}_A : A \to A$, $\mathrm{id}_B : B \to B$ are the identity morphisms) is the identity morphism $\mathrm{id}_{A‡B}$ for $A‡B$. In fact, for any morphisms $f \uplus g : A‡B \to C‡D$ (with $f : A \to C$, $g : B \to D$), $(f \uplus g) \circ (\mathrm{id}_A \uplus \mathrm{id}_B) = (f \circ \mathrm{id}_A) \uplus (g \circ \mathrm{id}_B) = f \uplus g$, and for any morphism $h \uplus k : C‡D \to A‡B$ (with $h : C \to A$, $k : D \to B$), $(\mathrm{id}_A \uplus \mathrm{id}_B) \circ (k \uplus h) = (\mathrm{id}_A \circ k) \uplus (\mathrm{id}_B \circ k) = h \uplus k$. Thus, $\mathrm{id} = (\mathrm{id}_A \uplus \mathrm{id}_B) : A‡B \to A‡B$ is equal to the identity arrow of $A‡B$, i. e.,
(2) $\mathrm{id}_A \uplus \mathrm{id}_B = \mathrm{id}_{A‡B}$.

In point 2 of Example 22, it was demonstrated that $(\mathrm{id}_A \uplus \mathrm{id}_B) : A \uplus B \to A \uplus B$ is the identity arrow $\mathrm{id}_{A \uplus B}$ of the object $A \uplus B$ as well. That is,
(3) $\mathrm{id}_A \uplus \mathrm{id}_B = \mathrm{id}_{A \uplus B}$.

Let us show now that the morphism $(\mathrm{id}_A \uplus \mathrm{id}_B) : A‡B \to A \uplus B$ is an isomorphism, with its inverse morphism $(\mathrm{id}_A \uplus \mathrm{id}_B)^{\mathrm{OP}} = \mathrm{id}_A^{\mathrm{OP}} \uplus \mathrm{id}_B^{\mathrm{OP}} = \mathrm{id}_A \uplus \mathrm{id}_B : A \uplus B \to A‡B$. In fact, $(\mathrm{id}_A \uplus \mathrm{id}_B) \circ (\mathrm{id}_A \uplus \mathrm{id}_B)^{\mathrm{OP}} = (\mathrm{id}_A \uplus \mathrm{id}_B) \circ (\mathrm{id}_A \uplus \mathrm{id}_B) = (\mathrm{id}_A \circ \mathrm{id}_A) \uplus (\mathrm{id}_B \circ \mathrm{id}_B) = \mathrm{id}_A \uplus \mathrm{id}_B =$ (from (3)) $= \mathrm{id}_{A \uplus B} : A \uplus B \to A \uplus B$ (the identity), and analogously, $(\mathrm{id}_A \uplus \mathrm{id}_B)^{\mathrm{OP}} \circ$

$(\mathrm{id}_A \biguplus \mathrm{id}_B) = \mathrm{id}_A \biguplus \mathrm{id}_B =$ (from (2)) $= \mathrm{id}_{A‡B} : A‡B \to A‡B$ (the identity). Consequently, $(\mathrm{id}_A \biguplus \mathrm{id}_B) : A‡B \to A \biguplus B$ is an isomorphism, i. e.,
(4) $A‡B \simeq A \biguplus B$.

Thus, we have from (4) and (1) that $A \biguplus B \simeq A‡B = B‡A \simeq B \biguplus A$, and hence, $A \biguplus B \simeq B \biguplus A$ as it was demonstrated in Lemma 3. □

Notice that $T(A‡B) = \{(\ddot{A}, v_i) \mid v_i \in TA\} \bigcup \{(\ddot{B}, v_j) \mid v_j \in TB\} = TA‡TB \neq T(A \biguplus B) = \{(1, v_i) \mid v_i \in TA\} \bigcup \{(2, v_j) \mid v_j \in TB\} = \widetilde{\mathrm{id}_A} \biguplus \widetilde{\mathrm{id}_B} = \widetilde{\mathrm{id}_A \biguplus \mathrm{id}_B} = \widetilde{\mathrm{id}_{A \biguplus B}}$. However, from (4), $\widetilde{\mathrm{id}_{A‡B}} = T(A‡B) \simeq T(A \biguplus B) = \widetilde{\mathrm{id}_{A \biguplus B}}$, so that for the identity morphism $\mathrm{id} : A‡B \to A‡B$ with $\widetilde{\mathrm{id}} = T(A‡B) \simeq \widetilde{\mathrm{id}_{A \biguplus B}}$, we obtain that $\mathrm{id} = \mathrm{id}_{A‡B}$ is a well-defined identity morphism.

Remark. From the fact that each object $A \biguplus B$, obtained by disjoint union, is isomorphic to the object $A‡B$, so that $A \biguplus B$ can be equivalently substituted by the separation-composed object $A‡B$, we will use only this second version of separation composition in the **DB** category (the disjoint union of objects in **DB** will be explicitly used prevalently for the demonstration purposes).

Corollary 21. *The following three isomorphisms for the separation composition of objects in* **DB** *are valid:*
- *(Composition with zero (initial and final) object)* $C‡ \perp^0 \simeq C \simeq \perp^0 ‡C$.
- *(Commutativity)* $A‡B \simeq B‡A$.
- *(Associativity)* $A‡(B‡C) \simeq A‡B‡C \simeq (A‡B)‡C$.

Proof. Claim 1. We have

$$C‡ \perp^0 \simeq C \biguplus \perp^0 \text{ (from Lemma 6)} \simeq C \text{ (from Lemma 3)}, \quad \text{and}$$
$$\perp^0 ‡C \simeq \perp^0 \biguplus C \text{ (from Lemma 6)} \simeq C \text{ (from Lemma 3)}.$$

For a given schema $\mathcal{C}$ with the interpretation α such that $C = \alpha^*(\mathcal{C})$, we have the identity morphism $\mathrm{id}_C = \alpha^*(\mathbf{M}_{CC}) : C \to C$ with $\mathbf{M}_{CC} = \alpha^*(\{1_r \mid r \in \mathcal{C}\} \bigcup \{1_{r_\emptyset}\})$.

Let us define, as in the proof of point 4 of Lemma 3, the morphisms $i_1 = \langle \alpha^*(\mathbf{M}_{CC}), \alpha^*(\{1_{r_\emptyset}\})\rangle = \langle \mathrm{id}_C, \perp^1\rangle : C \to C \biguplus \perp^0$, and $i_2 = [\alpha^*(\mathbf{M}_{CC}), \alpha^*(\{1_{r_\emptyset}\})] = [\mathrm{id}_C, \perp^1] : C \biguplus \perp^0 \to C$. Then, by using the isomorphism $\mathrm{id}_C \biguplus \perp^1 : C‡ \perp^0 \to C \biguplus \perp^0$, we obtain the composition $i_2 \circ (\mathrm{id}_C \biguplus \perp^1) = [\mathrm{id}_C, \perp^1] \circ (\mathrm{id}_C \biguplus \perp^1) = [\mathrm{id}_C \circ \mathrm{id}_C, \perp^1 \circ \perp^1] = [\mathrm{id}_C, \perp^1] = i_2 : C‡ \perp^0 \to C$, which is an isomorphism (it has been shown in the proof of point 4 of Lemma 3) and so is its inverse $(\mathrm{id}_C \biguplus \perp^1) \circ i_1 = i_1 : C \to C‡ \perp^0$.

Claim 2 (Commutativity).

$$A‡B \simeq A \biguplus B \text{ (from Lemma 6)} \simeq B \biguplus A \text{ (from point 4 of Lemma 3)}$$
$$\simeq B‡A \text{ (from Lemma 6)}.$$

Let us show that this commutativity is represented by the isomorphism

$$i_1 = \langle [\perp^1, \mathrm{id}_B], [\mathrm{id}_A, \perp^1] \rangle : A \ddagger B \to B \ddagger A,$$

where $[\perp^1, \mathrm{id}_B] : A \ddagger B \to B$, $[\mathrm{id}_A, \perp^1] : A \ddagger B \to A$.

In fact, in the proof of Lemma 3, we have shown that $i_1 : A \biguplus B \to B \biguplus A$ is an isomorphism. Let us show that $[\langle \mathrm{id}_A, \perp^1 \rangle, \langle \perp^1, \mathrm{id}_B \rangle] : A \ddagger B \to A \biguplus B$ is equal to the isomorphism $\mathrm{id}_A \biguplus \mathrm{id}_B : A \ddagger B \to A \biguplus B$. In fact, $\frac{[\langle \mathrm{id}_A, \perp^1 \rangle, \langle \perp^1, \mathrm{id}_B \rangle]}{\biguplus} = \{\mathrm{id}_A, \mathrm{id}_B\} = \frac{\mathrm{id}_A \biguplus \mathrm{id}_B}{\biguplus}$, and hence, from Definition 44, $[\langle \mathrm{id}_A, \perp^1 \rangle, \langle \perp^1, \mathrm{id}_B \rangle] = \mathrm{id}_A \biguplus \mathrm{id}_B$. Consequently, we can define the composition

$$\left(\mathrm{id}_B \biguplus \mathrm{id}_A\right) \circ i_1 \circ [\langle \mathrm{id}_A, \perp^1 \rangle, \langle \perp^1, \mathrm{id}_B \rangle] : A \ddagger B \to A \biguplus B \to B \biguplus A \to B \ddagger A,$$

and to show that it is equal to i_1 as follows:

$$\begin{aligned}
&\left(\mathrm{id}_B \biguplus \mathrm{id}_A\right) \circ i_1 \circ [\langle \mathrm{id}_A, \perp^1 \rangle, \langle \perp^1, \mathrm{id}_B \rangle] \\
&\quad = \left(\mathrm{id}_B \biguplus \mathrm{id}_A\right) \circ \langle [\perp^1, \mathrm{id}_B], [\mathrm{id}_A, \perp^1] \rangle \circ [\langle \mathrm{id}_A, \perp^1 \rangle, \langle \perp^1, \mathrm{id}_B \rangle] \\
&\quad = \langle \mathrm{id}_B \circ [\perp^1, \mathrm{id}_B], \mathrm{id}_A \circ [\mathrm{id}_A, \perp^1] \rangle \circ [\langle \mathrm{id}_A, \perp^1 \rangle, \langle \perp^1, \mathrm{id}_B \rangle] \\
&\quad = \langle [\perp^1, \mathrm{id}_B], [\mathrm{id}_A, \perp^1] \rangle \circ [\langle \mathrm{id}_A, \perp^1 \rangle, \langle \perp^1, \mathrm{id}_B \rangle] \quad \text{(from case 1 in Example 22)} \\
&\quad = \langle [[\perp^1, \mathrm{id}_B] \circ \langle \mathrm{id}_A, \perp^1 \rangle, [\perp^1, \mathrm{id}_B] \circ \langle \perp^1, \mathrm{id}_B \rangle], \\
&\qquad [[\mathrm{id}_A, \perp^1] \circ \langle \mathrm{id}_A, \perp^1 \rangle, [\mathrm{id}_A, \perp^1] \circ \langle \perp^1, \mathrm{id}_B \rangle] \rangle \\
&\quad = \langle [[\lceil \perp^1 \rfloor, \lceil \mathrm{id}_B \rfloor], [\lceil \mathrm{id}_A \rfloor, \lceil \perp^1 \rceil]] \rangle = \langle [\perp^1, \mathrm{id}_B], [\mathrm{id}_A, \perp^1] \rangle = i_1.
\end{aligned}$$

Claim 3 (Associativity).

$$\begin{aligned}
A \ddagger (B \ddagger C) &\simeq A \biguplus (B \ddagger C) \simeq A \biguplus \left(B \biguplus C\right) \quad \text{(from Lemma 6)} \\
&\simeq A \biguplus B \biguplus C = \biguplus (A, B, C) \quad \text{(from Lemma 3)} \\
&\simeq \ddagger (A, B, C) = A \ddagger B \ddagger C \quad \text{(from Lemma 6)}.
\end{aligned}$$

Similar to the proof above, $\mathrm{id}_A \biguplus \langle [\mathrm{id}_B, \perp^1], [\perp^1, \mathrm{id}_C] \rangle : A \ddagger (B \ddagger C) \to A \ddagger B \ddagger C$ represents this isomorphism. □

7.2.3 The (strong) behavioral equivalence for databases

Let us consider now the problem of defining the equivalent objects (instance-databases) from a *behavioral point of view based on observations*: as we have seen, each arrow (morphism) is composed of a number of "queries" (view maps) and each query may be considered as an *observation* over some database instance (i. e., an object of **DB**).

Thus, we can characterize each object in **DB** (i. e., an instance-database) by its behavior according to a given set of observations.

Indeed, if one object A is considered as a black-box, the object TA is the set of all observations on A. Thus, given two objects A and B, we are able to define the relation of equivalence between them based on the notion of the bisimulation relation. If the observations (i. e., resulting views of queries) of the instance-databases A and B are always equal, independently of their particular internal structure, then they look equivalent to an observer. In fact, any database can be seen as an Abstract Object Type (AOT) presented in the Appendix, Section A.3.2, by Definition 62, with a number of internal states that can be observed by using query operators (i. e., programs without side-effects). Consequently, two simple databases A and B are equivalent (or bisimilar) if they provide the same set of observations, i. e., when the set of views TA is equal to TB.

For the complex databases with separation operator ‡, represented equivalently (up to isomorphisms) by the nonsymmetric disjoint union $\biguplus$, the behavioral equivalence is more general relation than the isomorphism relation in **DB** category.

Definition 48. The relation of (strong) behavioral equivalence $'\approx'$ between objects (databases) in **DB** is defined by the PO relation '$\preceq$' (in Definition 40) on the set $\mathrm{Ob}_{\mathrm{DB}}$,

$$A \approx B \quad \text{iff} \quad A \preceq B \quad \text{and} \quad B \preceq A.$$

This equivalence relation for morphisms is defined by $f \approx g$ iff $\widetilde{f} \approx \widetilde{g}$.

Notice that for any arrow f in **DB**, $f \biguplus \perp^1 \approx \perp^1 \biguplus f \approx f$, where $\perp^1 = \alpha^*(\{1_{r_\emptyset}\}) = \{\alpha^*(1_{r_\emptyset})\} = \{q_\perp : \perp \to \perp\}$ is an empty morphism between objects (i. e., for any two objects A and B, we have an empty morphism $\perp^1 : A \to B$ with $\widetilde{\perp^1} = \{\perp\} = \perp^0$, obtained from the empty set of SOtgds between the schemas $\mathcal{A}$ and $\mathcal{B}$).

Moreover, any two simple arrows $f, g : A \to B$ are equal ($f \equiv g$, in Definition 44) iff they are observationally equivalent, i. e., if $f \approx g$.

This relation of behavioral equivalence between the objects does not strictly correspond to the notion of isomorphism in the category **DB** (see Proposition 25), in fact, we have the following.

Corollary 22. *Any two isomorphic objects A and B in* **DB** *are behaviorally equivalent, i. e., if $A \simeq B$, then $A \approx B$, but not vice versa.*

Only for the simple objects (databases) the behavioral equivalence corresponds to the isomorphism in **DB**.

Proof. If there is an isomorphism $A \simeq B$, with an iso-arrow is $: A \to B$ then, from the proof of Theorem 17, we obtain $A \preceq B$. Moreover, from duality, the arrow is$^{\mathrm{OP}} : B \to A$ is an isomorphism, and hence, $B \preceq A$. The converse does not hold: for example, $A \biguplus A \approx A$, but *not* $A \biguplus A \simeq A$. For two simple objects A and B, $A \preceq B$ means $TA \subseteq TB$. Thus, $A \approx B$ means $TA = TB$, so that $A \simeq TA = TB \simeq B$, i. e., $A \simeq B$. □

Notice that a database $A‡A$, which is composed of two copies of a database A, each one separated from another (each one has a different and independent DBMS), is behaviorally equivalent to a database A but not isomorphic to A. The behavioral equivalence is derived from the PO relation '$\preceq$' so that it will have an important role in the generation of the database complete lattice based on this ordering. It is well known that any two isomorphic objects in a given category can be mutually substituted, and from the fact that they are also behaviorally equivalent, it means that in this database lattice two isomorphic objects can be substituted as well by preserving PO ordering in this lattice.

7.2.4 Weak observational equivalence for databases

A database instance can also have some relations with tuples containing *Skolem constants*. For example, the minimal Herbrand models of a Global (virtual) schema of a data integration system with incomplete information [92, 98, 99] will have the Skolem constants for such a missed information.

We consider (see Definition 55 for FOL and Section A.3.1 in the introduction) a recursively enumerable set of all Skolem constants as marked (labeled) nulls $\mathrm{SK} = \{\omega_0, \omega_1, \ldots\} \subseteq \mathcal{U}$, disjoint from a database domain set $\mathbf{dom} \subseteq \mathcal{U}$ of all values for databases. Moreover, we introduce a unary predicate Val(_), such that $\mathrm{Val}(d_i)$ is true iff $d_i \in \mathbf{dom}$, and hence, $\mathrm{Val}(\omega_i)$ is false for any $\omega_i \in \mathrm{SK}$. Thus, we can define a new weak power-view operator for databases as follows.

Definition 49. Weak power-view operator $T_w : \mathrm{Ob}_{\mathrm{DB}} \longrightarrow \mathrm{Ob}_{\mathrm{DB}}$ is defined as follows: for any simple database A in a **DB** category,

$$T_w(A) \triangleq \{R \mid R \in T(A) \text{ and } \forall_{1 \le k \le |R|} \forall (d_i \in \pi_k(R)) \mathrm{Val}(d_i)\}$$

where $|R|$ is the number of attributes of the view (relation) R, and π_k is a k-th projection operator on relations. For $A = \biguplus_{1 \le j \le m} A_j$, $m \ge 1$, $T_w(A) = \biguplus_{1 \le j \le m} T_w(A_j)$.

We define a partial order relation '$\preceq_w$' for databases and a weak observational equivalence relation '$\approx_w$' (by $A \approx_w B$ iff $A \preceq_w B$ and $B \preceq_w A$) as in Definition 40 by substituting operator T by T_w.

The following properties are valid for the weak partial order $\preceq_w$, w. r. t. the partial order $\preceq$ (we denote $'A \prec B'$ iff $A \preceq B$ and not $A \approx B$).

Proposition 29. *Let $A = \biguplus_{1 \le j \le m} A_j$, $m \ge 1$ and $B = \biguplus_{1 \le i \le k} B_i$, $k \ge 1$ be any two databases (objects in* **DB** *category), then:*

1. *$T_w(A) \simeq A$, if A is a database without Skolem constants;*
   $T_w(A) \prec A$, otherwise;
2. *$A \preceq B$ implies $A \preceq_w B$, (thus, $A \approx B$ implies $A \approx_w B$);*
3. *$T_w(T_w(A)) = T(T_w(A)) = T_w(TA) = T_w(A) \subseteq TA$, thus, each object $D = T_w(A)$ is a closed object (i. e., $D = TD$) such that $D \approx_w A$.*

Proof. Claim 1. From $T_w(A) \subseteq TA$ ($T_w(A) = TA$, only if A is without Skolem constants).

Claim 2. If $A \prec B$, then $TA_j \subset TB_{\sigma(j)}$ with $T_w(TA_j) \subseteq T_w(TB_{\sigma(j)})$, for all $1 \leq j \leq m$, i. e., $A \preceq_w B$.

Claim 3. It holds from definition of the operator T and T_w: $T_w(T_w(A)) = T(T_w(A))$ because $T_w(A)$ is the set of views of A without Skolem constants and due to Claim 1. $T_w(TA) = \{R \mid R \in TT(A)$ and $\forall_{1\leq k\leq |R|}\forall(d_i \in \pi_k(R))\mathrm{Val}(d_i)\} = \{R \mid R \in TA$ and $\forall_{1\leq k\leq |R|}\forall(d_i \in \pi_k(R))\mathrm{Val}(d_i)\} = T_w(A)$, from $T = TT$. Let us show that $T_w(T_w(A)) = T_w(A)$. For every view $v \in T_w(T_w(A))$, from $T_w(T_w(A)) = T(T_w(A)) \subseteq TA$, it holds that $v \in TA$, and hence, from the fact that v is without Skolem constants, $v \in T_w(A)$. The converse is obvious. □

Notice that from point 2, the partial ordering '$\preceq$' is a stronger discriminator for databases than the weak partial order '$\preceq_w$', i. e., we can have two objects $A \prec B$ that are weakly equivalent, $A \approx_w B$ (e. g., when $A = T_w(B)$ and B is a database with Skolem constants). Let us extend the notion of the type operator T_w into the notion of the endofunctor of **DB** category.

Theorem 13. *There exists a weak power-view endofunctor $T_w = (T_w^0, T_w^1) : \mathbf{DB} \longrightarrow \mathbf{DB}$ such that the object component T_w^0 is equal to the type operator T_w and:*

1. *For any simple morphism $f : A \longrightarrow B$, the arrow component T_w^1 is defined by*

$$T_w(f) = T_w^1(f) \triangleq \{\mathrm{id}_R : R \to R \mid R \in \widetilde{f} \text{ and } \psi\}$$

where ψ is a statement $\forall_{1\leq k\leq |R|}\forall(d_i \in \pi_k(R))\mathrm{Val}(d_i)$, $|R|$ is the number of attributes of the relation (view) R, and π_k is a k-th projection. We define $T_w(f)$ for a complex arrow f by the set of ptp arrows $\frac{T_w(f)}{(+)} = \{T_w(f_{ji}) \mid f_{ji} \in \frac{f}{(+)}\}$.

2. *There exist the natural transformations $\xi : T_w \xrightarrow{\cdot} T$ (natural monomorphism) and $\xi^{-1} : T \xrightarrow{\cdot} T_w$ (natural epimorphism) such that for any simple object A, $\xi(A) = \mathrm{inc}_A$ is a monomorphism and $\xi^{-1}(A) = \mathrm{inc}_A^{\mathrm{OP}}$ is an epimorphism, which are equivalent, i. e., $\xi(A) \approx \xi^{-1}(A)$.*

Proof. Let us show that the morphism $T_w(f) : T_w(A) \to T_w(B)$, for a given morphism $f = \alpha^*(\mathbf{M}_{AB}) : A \to B$, corresponds to the definition of arrows in **DB** category, specified by Theorem 10, that is, there is a SOtgd Φ of a schema mapping such that $T_w(f) = \alpha^*(\mathrm{MakeOperads}(\{\Phi\}))$. Let us define the set of relational symbols in the instance-database $T_w(A)$ by $S_{T_wA} = \{r_i \mid r_i \in \mathbb{R}$ and $\alpha(r_i) \in T_w(A)\}$, and its schema $\mathcal{C} = (S_{T_wA}, \emptyset)$, so that $T_w(A) = \alpha^*(\mathcal{C})$ (for the instance-database $T_w(B)$, we define analogously the schema $\mathcal{D} = (S_{T_wB}, \emptyset)$). Let us define the set of relational symbols $S_{T_wf} = \{r_i \mid r_i \in \mathbb{R}$ and $R = \alpha(r_i) \in \widetilde{f}$ and $\forall_{1\leq k\leq |R|}\forall(d_i \in \pi_k(R))\mathrm{Val}(d_i)\}$. Then we define the SOtgd Φ by a formula $\bigwedge\{\forall \mathbf{x}_i(r_i(\mathbf{x}_i) \Rightarrow r_i(\mathbf{x}_i)) \mid r_i \in S_{T_wf}\}$.

In fact, $\mathrm{MakeOperads}(\{\Phi\}) = \{1_{r_i} \in O(r_i, r_i) \mid r_i \in S_{T_wf}\} \bigcup \{1_{r_\emptyset}\}$ is a sketch's mapping from the schema $\mathcal{C}$ into $\mathcal{D}$ and $T_w(f) = \alpha^*(\mathrm{MakeOperads}(\{\Phi\})) = \alpha^*(\{1_{r_i} \in O(r_i, r_i) \mid r_i \in S_{T_wf}\} \bigcup \{1_{r_\emptyset}\}) = \{\mathrm{id}_{r_i} : \alpha(r_i) \to \alpha(r_i) \in O(r_i, r_i) \mid r_i \in S_{T_wf}\} \bigcup \{q_\perp\}$.

For any identity arrow $\mathrm{id}_A : A \to A$, $T_w(\mathrm{id}_A) = \{\mathrm{id}_R : R \to R \mid R \in \widetilde{\mathrm{id}_A}$ and $\forall_{1\le k\le |R|}\forall(d_i \in \pi_k(R))\mathrm{Val}(d_i)\} = \{\mathrm{id}_R : R \to R \mid R \in TA$ and $\forall_{1\le k\le |R|}\forall(d_i \in \pi_k(R))\mathrm{Val}(d_i)\}$, with $\widetilde{T_w(\mathrm{id}_A)} = \{R \mid R \in TA$ and $\forall_{1\le k\le |R|}\forall(d_i \in \pi_k(R))\mathrm{Val}(d_i)\} = T_w(A)$ and hence we obtain the identity arrow $T_w(\mathrm{id}_A) = \mathrm{id}_{T_wA} : T_w(A) \to T_w(A)$. It is extended to identity arrows of complex objects analogously (the disjoint union of simple identity arrows).

It is easy to verify that for any two simple arrows $f : A \longrightarrow B$ and $g : B \longrightarrow C$,

$$\begin{aligned} T_w(g \circ f) &= \{\mathrm{id}_R : R \to R \mid R \in \widetilde{g \circ f} \text{ and } \forall_{1\le k\le |R|}\forall(d_i \in \pi_k(R))\mathrm{Val}(d_i)\} \\ &= \left\{\mathrm{id}_R : R \to R \mid R \in \widetilde{g} \bigcap \widetilde{f} \text{ and } \forall_{1\le k\le |R|}\forall(d_i \in \pi_k(R))\mathrm{Val}(d_i)\right\}. \end{aligned}$$

Thus,

$$\begin{aligned} \widetilde{T_w(g \circ f)} &= \left\{R \mid R \in \widetilde{g} \bigcap \widetilde{f} \text{ and } \forall_{1\le k\le |R|}\forall(d_i \in \pi_k(R))\mathrm{Val}(d_i)\right\} \\ &= \{R \mid R \in \widetilde{g} \text{ and } \forall_{1\le k\le |R|}\forall(d_i \in \pi_k(R))\mathrm{Val}(d_i)\} \\ &\quad \bigcap \{R \mid R \in \widetilde{f} \text{ and } \forall_{1\le k\le |R|}\forall(d_i \in \pi_k(R))\mathrm{Val}(d_i)\} \\ &= \widetilde{T_w(g)} \bigcap \widetilde{T_w(f)}. \end{aligned}$$

Consequently, $T_w(g \circ f) = T_w(g) \circ T_w(f)$. It holds for complex arrows as well (by considering the composition of their ptp arrows in $\frac{f}{\boxed{+}}$ and $\frac{g}{\boxed{+}}$).

Let us show that the monomorphism $\mathrm{inc}_A : T_w(A) \hookrightarrow TA$, for a given morphism $f = \alpha^*(\mathbf{M}_{AB}) : A \to B$, corresponds to the definition of arrows in **DB** category, specified by Theorem 10, i. e., there is a SOtgd Ψ of a schema mapping such that $\mathrm{inc}_A = \alpha^*(\mathrm{MakeOperads}(\{\Psi\}))$.

Let us define the set of relational symbols in the instance-database $T_w(A)$ by $S_{T_wA} = \{r_i \mid r_i \in \mathbb{R}$ and $\alpha(r_i) \in T_w(A)\}$, and its schema $\mathcal{C} = (S_{T_wA}, \emptyset)$, so that $T_w(A) = \alpha^*(\mathcal{C})$ (we define analogously the schema $T\mathcal{A} = (S_{TA}, \emptyset)$ for the instance-database TA). Then we define the SOtgd Φ by a conjunctive formula $\bigwedge\{\forall \mathbf{x}_i(r_i(\mathbf{x}_i) \Rightarrow r_i(\mathbf{x}_i)) \mid r_i \in S_{T_wA}\}$.

In fact, $\mathrm{MakeOperads}(\{\Psi\}) = \{1_{r_i} \in O(r_i, r_i) \mid r_i \in S_{T_wA}\} \bigcup \{1_{r_\emptyset}\}$ is a sketch's mapping from the schema $\mathcal{C}$ into $T\mathcal{A}$ and $\mathrm{inc}_A = \alpha^*(\mathrm{MakeOperads}(\{\Psi\})) = \alpha^*(\{1_{r_i} \in O(r_i, r_i) \mid r_i \in S_{T_wA}\} \bigcup \{1_{r_\emptyset}\}) = \{\mathrm{id}_{r_i} : \alpha(r_i) \to \alpha(r_i) \in O(r_i, r_i) \mid r_i \in S_{T_wA}\} \bigcup \{q_\perp\}$.

Thus, $\widetilde{\mathrm{inc}_A} = T\{\alpha(r_i) \in O(r_i, r_i) \mid r_i \in S_{T_wA}\} = T(T_w(A)) =$ (from point 4 of Proposition 29) $= T_w(A)$, and hence, $\mathrm{inc}_A : T_w(A) \hookrightarrow TA$ is a monomorphism (for the inclusion $T_w(A) \subseteq TA$) and, by duality, $\mathrm{inc}_A^{\mathrm{OP}} : TA \twoheadrightarrow T_w(A)$ is an epimorphism. It is easy to verify that the natural transformations ξ and ξ^{-1} are well-defined. □

Analogously to the monad (T, η, μ) and comonad (T, η^C, μ^C) of the endofunctor T, we can define such structures for the weak endofunctor T_w as well.

Proposition 30. *The weak power-view endofunctor $T_w = (T_w^0, T_w^1) : \mathbf{DB} \longrightarrow \mathbf{DB}$ defines the monad (T_w, η_w, μ_w) and the comonad (T_w, η_w^C, μ_w^C) in* **DB** *such that $\eta_w = \xi^{-1} \bullet \eta : I_{\mathrm{DB}} \xrightarrow{\cdot} T_w$ is a natural epimorphism and $\eta_w^C = \eta^C \bullet \xi : T_w \xrightarrow{\cdot} I_{\mathrm{DB}}$ is a natural*

monomorphism ('•' is a vertical composition for natural transformations), while $\mu_w : T_wT_w \xrightarrow{\cdot} T_w$ *and* $\mu_w^C : T_w \xrightarrow{\cdot} T_wT_w$ *are equal to the natural identity transformation* $\mathrm{id}_{T_w} : T_w \xrightarrow{\cdot} T_w$ *(because* $T_w = T_wT_w$*).*

Proof. It is easy to verify that all commutative diagrams of the monad and the comonad are composed of identity arrows. □

7.3 Internal categorial symmetry of DB category

The mapping $B_\top : \mathrm{Mor}_{\mathrm{DB}} \longrightarrow \mathrm{Ob}_{\mathrm{DB}}$, specified in definition of the **DB** category in Theorem 10, is a fundamental concept for the conceptual categorial symmetry [4] introduced in Section 3.3. Let us demonstrate the following equality for any morphism in **DB**, which will be used in the proof of the next theorem.

Lemma 7. *For any morphism* $f : A \to B$, $f = f \circ f^{\mathrm{OP}} \circ f$. *Let* $\mathrm{in}_{\widetilde{f}} = \mathrm{in}_B : \widetilde{f} \hookrightarrow TB$ *and* $\mathrm{in}_{\widetilde{f^{\mathrm{OP}}}} = \mathrm{in}_A : \widetilde{f} \hookrightarrow TA$ *be two monomorphisms defined by Proposition* 24.

Then $\mathrm{in}_{\widetilde{f^{\mathrm{OP}}}} \circ \mathrm{in}_{\widetilde{f}}^{\mathrm{OP}} = Tf^{\mathrm{OP}} : TB \to TA$.

Proof. If f is a simple arrow, then $\widetilde{f} = \widetilde{f^{\mathrm{OP}}}$ (by duality), so that $f \circ \widetilde{f^{\mathrm{OP}}} \circ f = \widetilde{f} \bigcap \widetilde{f^{\mathrm{OP}}} \bigcap \widetilde{f} = \widetilde{f}$, and hence, from Definition 44,
(1) $f = f \circ f^{\mathrm{OP}} \circ f$.

For any two given objects $A = \biguplus_{1\le j\le m} A_j$, $m \ge 1$ and $B = \biguplus_{1\le i\le k} B_i$, $k \ge 1$ and a complex arrow f between them, $\frac{f^{\mathrm{OP}}}{\biguplus} = \{f_{ji}^{\mathrm{OP}} \mid f_{ji} \in \frac{f}{\biguplus}\}$, so that

$$\frac{f^{\mathrm{OP}} \circ f}{\biguplus} = \left\{ f_{ji}^{\mathrm{OP}} \circ f_{ji} : A_j \to A_j \mid f_{ji} : A_j \to B_i \in \frac{f}{\biguplus} \right\}.$$

Thus,

$$\begin{aligned}
\frac{f \circ f^{\mathrm{OP}} \circ f}{\biguplus} &= \left\{ f_{ji} \circ h : A_j \to B_i \mid f_{ji} : A_j \to B_i \in \frac{f}{\biguplus}, \mathrm{dom}(f_{ji}) = A_j = \mathrm{cod}(h), h \in \frac{f^{\mathrm{OP}} \circ f}{\biguplus} \right\} \\
&= \left\{ f_{ji} \circ (f_{ji}^{\mathrm{OP}} \circ f_{ji}) : A_j \to B_i \mid f_{ji} : A_j \to B_i \in \frac{f}{\biguplus} \right\} \\
&= \left\{ f_{ji} : A_j \to B_i \mid f_{ji} \in \frac{f}{\biguplus} \right\} \quad \text{(from (1) and fact that } f_{ji} \text{ are simple arrows)} \\
&= \frac{f}{\biguplus}.
\end{aligned}$$

So, from Definition 44, $f = f \circ f^{\mathrm{OP}} \circ f$ is valid for complex morphisms as well.

From point 1 of Proposition 24, we have for any morphism $f : A \to B$ the monomorphisms $\mathrm{in}_{\widetilde{f}} = \mathrm{in}_B : \widetilde{f} \hookrightarrow TB$ and $\mathrm{in}_{\widetilde{f^{\mathrm{OP}}}} = \mathrm{in}_A : \widetilde{f} \hookrightarrow TA$ with $\frac{\mathrm{in}_A}{\biguplus} = \{\mathrm{in}_{ji} = \{\mathrm{id}_R \mid R \in \widetilde{f_{ji}}\} :$

$\widetilde{f_{ji}} \hookrightarrow TA_j \mid (f_{ji} : A_j \to B_i) \in \frac{f}{\lfloor + \rfloor}\}$, $\frac{\mathrm{in}_B}{\lfloor + \rfloor} = \{\mathrm{in}_{ji} = \{\mathrm{id}_R \mid R \in \widetilde{f_{ji}}\} : \widetilde{f_{ji}} \hookrightarrow TB_i \mid (f_{ji} : A_j \to B_i) \in \frac{f}{\lfloor + \rfloor}\}$. Thus,

$$\frac{\mathrm{in}_{\widetilde{f}}^{\mathrm{OP}}}{\lfloor + \rfloor} = \left\{\mathrm{in}_{ji}^{\mathrm{OP}} = \{\mathrm{id}_R \mid R \in \widetilde{f_{ji}}\} : TB_i \twoheadrightarrow \widetilde{f_{ji}} \mid (f_{ji} : A_j \to B_i) \in \frac{f}{\lfloor + \rfloor}\right\} \quad \text{and}$$

$$\frac{\mathrm{in}_{\widetilde{f^{\mathrm{OP}}}} \circ \mathrm{in}_{\widetilde{f}}^{\mathrm{OP}}}{\lfloor + \rfloor} = \left\{\mathrm{in}_{ji}^{\mathrm{OP}} \circ \mathrm{in}_{ji} = \{\mathrm{id}_R \circ \mathrm{id}_R \mid R \in \widetilde{f_{ji}}\} : TB_i \twoheadrightarrow f_{ji} \hookrightarrow TA_j \mid (f_{ji} : A_j \to B_i) \in \frac{f}{\lfloor + \rfloor}\right\}$$

$$= \left\{\{\mathrm{id}_R \mid R \in \widetilde{f_{ji}}\} : TB_i \to TA_j \mid (f_{ji} : A_j \to B_i) \in \frac{f}{\lfloor + \rfloor}\right\} \quad \text{(from Theorem 12)}$$

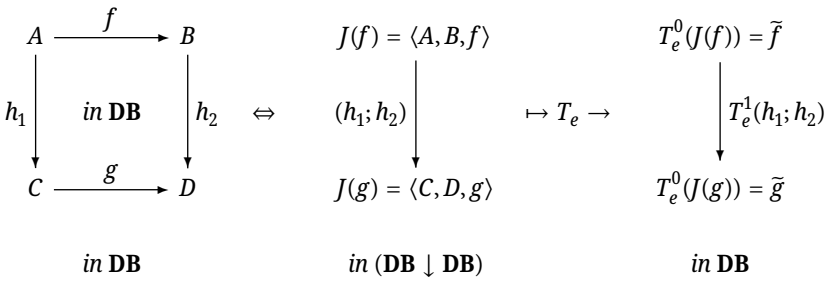
$$= \left\{Tf_{ji}^{\mathrm{OP}} : TB_i \to TA_j \mid (f_{ji} : A_j \to B_i) \in \frac{f}{\lfloor + \rfloor}\right\} = \frac{Tf^{\mathrm{OP}}}{\lfloor + \rfloor}.$$

Consequently, from Definition 44, we obtain $\mathrm{in}_{\widetilde{f^{\mathrm{OP}}}} \circ \mathrm{in}_{\widetilde{f}}^{\mathrm{OP}} = Tf^{\mathrm{OP}} : TB \to TA$. □

Now we are ready to demonstrate that the **DB** category is a symmetry-extended category, w. r. t. Definition 14 in Section 3.4, with introductions of the encapsulation operator J such that for each morphisms $f : A \to B$ in **DB**, $J(f) = \langle A, B, f\rangle$ is an object in the arrow (or comma) category **DB** ↓ **DB** with its inverse operator ψ such that $\psi(J(f)) = f$.

Theorem 14. *The category* **DB** *is a symmetry-extended category, closed by the functor* $T_e = (T_e^0, T_e^1) : (\mathbf{DB} \downarrow \mathbf{DB}) \longrightarrow \mathbf{DB}$*, where* $T_e^0 = B_\top\psi$ *is the object component of this functor such that for any arrow* $f : A \to B$ *in* **DB**, $T_e^0(J(f)) = \widetilde{f}$*, while its arrow component* T_e^1 *is defined as follows:*

For any arrow $(h_1; h_2) : J(f) \longrightarrow J(g)$ *in* **DB** ↓ **DB** *(so that* $g \circ h_1 = h_2 \circ f$ *in* **DB)**, $T_e^1(h_1; h_2) \triangleq \mathrm{in}_{\widetilde{g}}^{\mathrm{OP}} \circ T(h_2 \circ f) \circ \mathrm{in}_{\widetilde{f^{\mathrm{OP}}}}$*, where* $\mathrm{in}_{\widetilde{f^{\mathrm{OP}}}} : \widetilde{f} \hookrightarrow TA$ *and* $\mathrm{in}_{\widetilde{g}} : \widetilde{g} \hookrightarrow TD$ *are two monomorphisms (Proposition 24). This is represented by*



The associative composition operator for objects $*$*, defined for any fitted pair* $g \circ f$ *of arrows, satisfies* $B_\top(g) * B_\top(f) \triangleq \widetilde{g \circ f} = B_\top(g \circ f)$.

Proof. Let us consider a simple morphism $f = \alpha^*(\mathbf{M}_{AB}) : A \to B$. In accordance with Theorem 10, for $\mathbf{M}_{AB} = \mathrm{MakeOperads}(\mathcal{M}_{AB})$, $A = \alpha^*(\mathcal{A})$, $B = \alpha^*(\mathcal{B})$ with $(\Phi_E, \Phi_M) = \mathrm{DeCompose}(\mathcal{M}_{AB})$, and let us define the mapping-operads $\mathbf{M}_E = \mathrm{MakeOperads}(\{\Phi_E\})$ and $\mathbf{M}_M = \mathrm{MakeOperads}(\{\Phi_M\})$.

Let us show that the monomorphism $\mathrm{in}_{\widetilde{f^{\mathrm{OP}}}} = \mathrm{in}_A : \widetilde{f} \hookrightarrow TA$ (from Lemma 7), for this morphisms $f = \alpha^*(\mathrm{MakeOperads}(\mathcal{M}_{AB})) : A \to B$, corresponds to the definition of arrows in **DB** category, specified by Theorem 10, i. e., there exists an SOtgd Φ of a schema mapping such that $\mathrm{in}_{\widetilde{f^{\mathrm{OP}}}} = \alpha^*(\mathrm{MakeOperads}(\{\Phi\}))$.

Let us define the set of relational symbols $S = \{r_i \mid r_i \in \mathbb{R} \text{ and } \alpha(r_i) \in \Delta(\alpha, \mathbf{M}_{AB}) \subseteq \widetilde{f}\} \subseteq S_{\widetilde{f}} = \{r_i \mid r_i \in \mathbb{R} \text{ and } \alpha(r_i) \in \widetilde{f}\}$ with $\widetilde{f} = TS$ and its schema $\mathcal{C} = (S_{\widetilde{f}}, \emptyset)$, so that $\widetilde{f} = \alpha^*(\mathcal{C})$ (analogously, from Definition 70, for the instance-database TA we define the schema $(S_{TA}, \emptyset) = \mathcal{A} \bigsqcup_{\alpha} \mathcal{A}$). Consequently, we define the SOtgd Φ by a conjunctive formula $\bigwedge\{\forall \mathbf{x}_i (r_i(\mathbf{x}_i) \Rightarrow r_i(\mathbf{x}_i)) \mid r_i \in S\}$.

In fact, $\mathrm{MakeOperads}(\{\Phi\}) = \{1_{r_i} \in O(r_i, r_i) \mid r_i \in S\} \bigcup \{1_{r_\emptyset}\}$ is a sketch's mapping between schemas $\mathcal{C}$ and $T\mathcal{A}$. Consequently,

$$\begin{aligned}\mathrm{in}_{\widetilde{f^{\mathrm{OP}}}} &= \alpha^*(\mathrm{MakeOperads}(\{\Phi\})) = \alpha^*\left(\{1_{r_i} \in O(r_i, r_i) \mid r_i \in S\} \bigcup \{1_{r_\emptyset}\}\right) \\ &= \left(\{\mathrm{id}_R = \alpha(1_{r_i}) : R \to R \mid R = \alpha(r_i) \in S \subseteq \widetilde{f}\} \bigcup \{q_\perp\}\right) : \widetilde{f} \hookrightarrow TA,\end{aligned}$$

with $\widetilde{\mathrm{in}_{\widetilde{f^{\mathrm{OP}}}}} = \widetilde{f}$.

Thus, from the duality theorem, this property holds for the epimorphism $\mathrm{in}^{\mathrm{OP}}_{\widetilde{f}} : TB \twoheadrightarrow \widetilde{f}$ (which is a dual (inverted arrow) of the monomorphism $\mathrm{in}_{\widetilde{f}} = \mathrm{in}_B : \widetilde{f} \hookrightarrow TB$ in Lemma 7) as well.

Thus, the morphism $T^1_e(h_1; h_2) = \mathrm{in}^{\mathrm{OP}}_{\widetilde{g}} \circ T(h_2 \circ f) \circ \mathrm{in}_{\widetilde{f^{\mathrm{OP}}}}$ is well-defined in **DB** due to Theorem 10. It is easy to verify that also T_e is a well-defined functor. In fact, for any identity arrow $(\mathrm{id}_A; \mathrm{id}_B) : J(f) \longrightarrow J(f)$, $T^1_e(\mathrm{id}_A; \mathrm{id}_B) = \mathrm{in}^{\mathrm{OP}}_{\widetilde{f}} \circ T(\mathrm{id}_B \circ f) \circ \mathrm{in}_{\widetilde{f^{\mathrm{OP}}}} = \mathrm{in}^{\mathrm{OP}}_{\widetilde{f}} \circ Tf \circ \mathrm{in}_{\widetilde{f^{\mathrm{OP}}}}$. Thus, when $f, \mathrm{id}_A, \mathrm{id}_B$ are simple arrows, $\widetilde{T^1_e(\mathrm{id}_A; \mathrm{id}_B)} = \widetilde{\mathrm{in}^{\mathrm{OP}}_{\widetilde{f}}} \bigcap \widetilde{Tf} \bigcap \widetilde{\mathrm{in}_{\widetilde{f^{\mathrm{OP}}}}} = \widetilde{f}$ (from $\widetilde{\mathrm{in}^{\mathrm{OP}}_{\widetilde{f}}} = \widetilde{Tf} = \widetilde{\mathrm{in}_{\widetilde{f^{\mathrm{OP}}}}} = \widetilde{f}$) and, consequently, $T^1_e(\mathrm{id}_A; \mathrm{id}_B) : \widetilde{f} \to \widetilde{f}$ is equal to the identity arrow $\mathrm{id}_{\widetilde{f}}$. It is easy to verify that it holds also when f, id_A and id_B are complex arrows as well.

For any two arrows $(h_1; h_2) : J(f) \longrightarrow J(g)$ and $(l_1; l_2) : J(g) \longrightarrow J(k)$ if all arrows f, g, k, h_1, h_2, l_1 and l_2 are simple arrows then

$$\begin{aligned}\overline{T^1_e(l_1; l_2) \circ T^1_e(h_1; h_2)} &= \widetilde{T^1_e(l_1; l_2)} \bigcap \widetilde{T^1_e(h_1; h_2)} \\ &= \left(\widetilde{k} \bigcap \widetilde{l_2} \bigcap \widetilde{g}\right) \bigcap \left(\widetilde{g} \bigcap \widetilde{h_2} \bigcap \widetilde{f}\right) \\ &= \widetilde{k} \bigcap \widetilde{l_2} \bigcap \widetilde{g} \bigcap \widetilde{h_2} \bigcap \widetilde{f} \quad (\text{by } h_2 \circ f = g \circ h_1) \\ &= \widetilde{k} \bigcap \widetilde{l_2} \bigcap \widetilde{g} \bigcap \widetilde{h_1} \bigcap \widetilde{h_2} \bigcap \widetilde{f} \quad (\text{by } h_2 \circ f = g \circ h_1) \\ &= \widetilde{k} \bigcap \widetilde{l_2} \bigcap \widetilde{h_2} \bigcap \widetilde{f} = \widetilde{k} \bigcap \widetilde{l_2 \circ h_2} \bigcap \widetilde{f} \\ &= \overline{\mathrm{in}^{\mathrm{OP}}_{\widetilde{k}} \circ T((l_2 \circ h_2) \circ f) \circ \mathrm{in}_{\widetilde{f^{\mathrm{OP}}}}}\end{aligned}$$

$$= T_e^1(l_1 \circ \widetilde{h_1; l_2} \circ h_2)$$
$$= T_e^1(\widetilde{(l_1; l_2)} \circ (h_1; h_2)).$$

Thus, $T_e^1((l_1; l_2) \circ (h_1; h_2)) = T_e^1(l_1; l_2) \circ T_e^1(h_1; h_2)$.

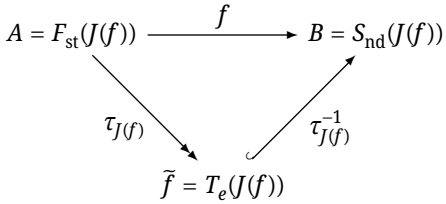
Let us show this functorial composition property also when we have the complex arrows. We have (from Lemma 7):

(1) $\mathrm{in}^{\mathrm{OP}}_{\widetilde{g^{\mathrm{OP}}}} \circ \mathrm{in}^{\mathrm{OP}}_{\widetilde{g}} = Tg^{\mathrm{OP}}$. Thus,

$$\begin{aligned}
T_e^1(l_1; l_2) \circ T_e^1(h_1; h_2) &= \mathrm{in}^{\mathrm{OP}}_{\widetilde{k}} \circ T(l_2 \circ g) \circ \mathrm{in}_{\widetilde{g^{\mathrm{OP}}}} \circ \mathrm{in}^{\mathrm{OP}}_{\widetilde{g}} \circ T(h_2 \circ f) \circ \mathrm{in}_{\widetilde{f^{\mathrm{OP}}}} \\
&= \mathrm{in}^{\mathrm{OP}}_{\widetilde{k}} \circ T(l_2 \circ g) \circ Tg^{\mathrm{OP}} \circ T(h_2 \circ f) \circ \mathrm{in}_{\widetilde{f^{\mathrm{OP}}}} \quad \text{(from (1))} \\
&= \mathrm{in}^{\mathrm{OP}}_{\widetilde{k}} \circ T(l_2 \circ g \circ g^{\mathrm{OP}} \circ h_2 \circ f) \circ \mathrm{in}_{\widetilde{f^{\mathrm{OP}}}} \\
&= \mathrm{in}^{\mathrm{OP}}_{\widetilde{k}} \circ T(l_2 \circ (g \circ g^{\mathrm{OP}} \circ g) \circ h_1) \circ \mathrm{in}_{\widetilde{f^{\mathrm{OP}}}} \quad \text{(from } h_2 \circ f = g \circ h_1\text{)} \\
&= \mathrm{in}^{\mathrm{OP}}_{\widetilde{k}} \circ T(l_2 \circ g \circ h_1) \circ \mathrm{in}_{\widetilde{f^{\mathrm{OP}}}} \quad \text{(from Lemma 7)} \\
&= \mathrm{in}^{\mathrm{OP}}_{\widetilde{k}} \circ T((l_2 \circ h_2) \circ f) \circ \mathrm{in}_{\widetilde{f^{\mathrm{OP}}}} \quad \text{(from } h_2 \circ f = g \circ h_1\text{)} \\
&= T_e^1(l_1 \circ h_1; l_2 \circ h_2) = T_e^1((l_1; l_2) \circ (h_1; h_2)).
\end{aligned}$$

For any identity arrow id_A, $T_e^0 J(\mathrm{id}_A) = \widetilde{\mathrm{id}_A} = TA \simeq A$. Thus, as required by Definition 7, the natural isomorphism $\varphi : T_e \circ \blacktriangle \simeq I_{\mathrm{DB}}$ is valid.

Let us show that **DB** is an *symmetry-extended* category, i. e., $\tau^{-1} \bullet \tau = \psi : F_{\mathrm{st}} \xrightarrow{\cdot} S_{\mathrm{nd}}$ for vertical composition of natural transformations $\tau : F_{\mathrm{st}} \xrightarrow{\cdot} T_e$ and $\tau^{-1} : T_e \xrightarrow{\cdot} S_{\mathrm{nd}}$, so that for each object $J(f) = \langle A, B, f \rangle$ in **DB** $\downarrow$ **DB**, the following diagram commutes (we recall that $\psi_{J(f)} = \psi(J(f)) = f$):



where $T_e(J(f)) = \widetilde{f} \simeq \widetilde{\biguplus \frac{f}{\biguplus}} = \biguplus_{f_{ji} \in \frac{f}{\biguplus}} \widetilde{f_{ji}}$ and $\frac{\tau_{J(f)}}{\biguplus} = \{\tau_{J(f_{ji})} : A_j \twoheadrightarrow \widetilde{f_{ji}} \mid (f_{ji} : A_j \to B_i) \in \frac{f}{\biguplus}\}$ and $\frac{\tau^{-1}_{J(f)}}{\biguplus} = \{\tau^{-1}_{J(f_{ji})} : \widetilde{f_{ji}} \hookrightarrow B_i \mid (f_{ji} : A_j \to B_i) \in \frac{f}{\biguplus}\}$.

In fact, for a simple morphism $f = \alpha^*(\mathrm{MakeOperads}(\mathcal{M}_{AB})) : A \to B$, we have two instance-mappings $\mathrm{ep}(f) = \alpha^*(\mathbf{M}_E) : A \to TC$ and $in(f) = \alpha^*(\mathbf{M}_M) : TC \to B$ with $\mathcal{C} = (S_C, \emptyset)$, $S_C = \bigcup_{q'_i \in \mathbf{M}_E} \partial_1(q'_i) = \{r_{q_1}, \ldots, r_{q_m}\}$ and $C = \alpha^*(S_C)$. From the fact that $TC = T\alpha^*(S_C) \subseteq \widetilde{f}$, we can define the equivalent morphisms to these two morphisms, by substituting the instance-database TC by $\widetilde{f}$, and hence, defining the epimorphism $\tau_{J(f)} \triangleq$

$\alpha^*(\mathbf{M}_E) : A \twoheadrightarrow \widetilde{f}$ ($\widetilde{\tau_{J(f)}} = \widetilde{\mathrm{ep}(f)} = \widetilde{f}$) and the monomorphism $\tau^{-1}_{J(f)} \triangleq \alpha^*(\mathbf{M}_M) : \widetilde{f} \hookrightarrow B$ ($\widetilde{\tau^{-1}_{J(f)}} = \widetilde{in(f)} = \widetilde{f}$).

Notice that the existence of this monomorphism is also predicted by Proposition 24 with monic "inclusion" $\widetilde{f} \preceq TB$ (i. e., $\mathrm{in}_B : \widetilde{f} \hookrightarrow TB$) and isomorphism $\mathrm{is}^{-1}_B : TB \simeq B$, which is monic as well, and hence, their composition is a monomorphism equal to $\tau^{-1}_{J(f)} = \mathrm{is}^{-1}_B \circ \mathrm{in}_B : \widetilde{f} \hookrightarrow B$. In the same way, from the epimorphism $\mathrm{in}^{\mathrm{OP}}_A : TA \twoheadrightarrow \widetilde{f}$ (the dual (inverted) of the monomorphism $\mathrm{in}_A : \widetilde{f} \hookrightarrow TA$) and isomorphism $\mathrm{is}_A : A \simeq TA$, which is epic as well, we obtain that their composition is epic as well. That is, $\tau_{J(f)} = \mathrm{in}^{\mathrm{OP}}_A \circ \mathrm{is}_A : A \twoheadrightarrow \widetilde{f}$ is an epimorphism. □

Remark. For any given morphism $f : A \longrightarrow B$ in **DB**, the arrow $f_{\mathrm{ep}} = \tau(J(f)) : A \twoheadrightarrow \widetilde{f}$ is an epimorphism, and the arrow $f_{in} = \tau^{-1}(J(f)) : \widetilde{f} \hookrightarrow B$ is a monomorphism, so that *any* morphism f in **DB** is a composition of *an epimorphism and monomorphism* $f = f_{in} \circ f_{\mathrm{ep}}$ (as in **Set**, in Example 10), with the intermediate object equal to its information flux $\widetilde{f}$.

7.3.1 (Co)products

We recall that for a tuple of objects (i. e., instance-databases) in **DB** category $(S_1, \ldots, S_n)$, $n \geq 2$, we define its disjoint union (see Definition 40),

$$\biguplus_{1 \leq i \leq n} S_i = \biguplus(S_1, \ldots, S_n) = \bigcup_{1 \leq i \leq n} \{(i, a) \mid a \in S_i\}$$

the union *indexed by positions* of the sets in a given tuple, denoted also as $S_1 \uplus S_2 \uplus \cdots \uplus S_n$. It will be demonstrated that the disjoint union of two objects is isomorphic to separation composition of these objects.

Theorem 15. *There exists an idempotent coproduct bifunctor* $+ : \mathbf{DB} \times \mathbf{DB} \longrightarrow \mathbf{DB}$, *which is a disjoint union* $\uplus$ *for objects and arrows in* **DB**.

The category **DB** *is co-Cartesian with initial (and terminal) object* $\perp^0$ *and for every pair of objects A and B, the object* $A + B$ *is a coproduct with monomorphisms (injections)* $\mathrm{in}_A = \langle \mathrm{id}_A, \perp^1 \rangle : A \hookrightarrow A + B$ *and* $\mathrm{in}_B = \langle \perp^1, \mathrm{id}_B \rangle : B \hookrightarrow A + B$.

Based on duality property, **DB** *is also Cartesian category with a terminal object* $\perp^0$. *For each pair of objects A and B, there exists a categorial product* $A \times B$ *with epimorphisms (projections)* $p_A = \mathrm{in}^{\mathrm{OP}}_A : A \times B \twoheadrightarrow A$ *and* $p_B = \mathrm{in}^{\mathrm{OP}}_B : A \times B \twoheadrightarrow B$.

Thus, the product bifunctor is equal to the coproduct bifunctor, i. e., $\times \equiv +$.

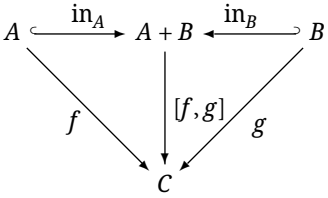
Proof. Note that the coproduct + is just the disjoint union $\uplus$, so that:

1. For any identity arrow $(\mathrm{id}_A, \mathrm{id}_B)$ in $\mathbf{DB} \times \mathbf{DB}$ where id_A and id_B are the identity arrows of A and B, respectively, $\frac{\mathrm{id}_A + \mathrm{id}_B}{\uplus} = \frac{\mathrm{id}_{A+B}}{\uplus} = \{\mathrm{id}_A, \mathrm{id}_B\}$. Thus, $+^1(\mathrm{id}_A, \mathrm{id}_B) = \mathrm{id}_A + \mathrm{id}_B = \mathrm{id}_{A+B}$ is an identity arrow of the object $A + B$.

2. For any $k : A \longrightarrow A_1$, $k_1 : A_1 \longrightarrow A_2$, $l : B \longrightarrow B_1$, $l_1 : B_1 \longrightarrow B_2$, it holds $\frac{+^1(k_1,l_1)\circ+^1(k,l)}{\biguplus} = \{k_1 \circ k : A \to A_2, l_1 \circ l : B \to B_2\} = \frac{+^1(k_1\circ k, l_1\circ l)}{\biguplus}$. Thus, from point 3 of Definition 44, it is valid functorial property for composition of morphisms $+^1(k_1\circ k, l_1\circ l) = +^1(k_1,l_1)\circ+^1(k,l)$.

3. Let us demonstrate the coproduct property of this bifunctor. For any two simple arrows $f : A \longrightarrow C$ and $g : B \longrightarrow C$ there exists a unique arrow $k = [f,g] : A + B \longrightarrow C$ such that $f = k \circ \mathrm{in}_A$ and $g = k \circ \mathrm{in}_B$ where $\mathrm{in}_A = \langle \mathrm{id}_A, \perp^1\rangle : A \hookrightarrow A + B$ and $\mathrm{in}_B = \langle \perp^1, \mathrm{id}_B\rangle : B \hookrightarrow A + B$ are the monomorphisms (injections). In fact, for any two arrows $h,k : D \to A$, if $\mathrm{in}_A \circ h = \mathrm{in}_A \circ k$ then $h = k$ ($\mathrm{in}_A \circ h = \langle \mathrm{id}_A, \perp^1\rangle \circ h = \langle \mathrm{id}_A \circ h, \perp^1 \circ h\rangle = \langle h, \perp^1\rangle = \mathrm{in}_A \circ k = \langle h, \perp^1\rangle$, so that $h = k$). Analogously, it holds also for in_B. That is, the following coproduct diagram in **DB** commutes:

$+ \equiv \biguplus$,



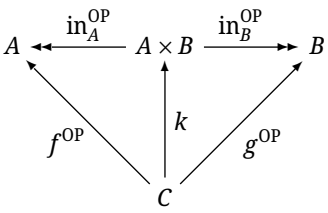
Assume that, for a given schema $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$, we have the sketch's maggings $\mathbf{M}_{AC} : \mathcal{A} \to \mathcal{C}$ and $\mathbf{M}_{BC} : \mathcal{B} \to \mathcal{C}$ and an interpretation α such that $A = \alpha^*(\mathcal{A})$, $B = \alpha^*(\mathcal{B})$, $C = \alpha^*(\mathcal{C})$, $f = \alpha^*(\mathbf{M}_{AC})$ and $g = \alpha^*(\mathbf{M}_{BC})$. Then we define the morphisms $\mathrm{in}_A = \alpha^*(\langle \mathbf{M}_{AA}, \{1_{r_\emptyset}\}\rangle) = \langle \mathrm{id}_A, \perp^1\rangle$ and $\mathrm{in}_B = \alpha^*(\langle \{1_{r_\emptyset}\}, \mathbf{M}_{BB}\rangle) = \langle \perp^1, \mathrm{id}_B\rangle$ where $\mathbf{M}_{AA} = \{1_r \mid r \in \mathcal{A}\} \bigcup \{1_{r_\emptyset}\}$ and $\mathbf{M}_{BB} = \{1_r \mid r \in \mathcal{B}\} \bigcup \{1_{r_\emptyset}\}$.

Thus, $[f,g] \circ \mathrm{in}_A = [f,g] \circ \langle \mathrm{id}_A, \perp^1\rangle = \lceil f \circ \mathrm{id}_A, g \circ \perp^1\rfloor = \lceil f, \perp^1\rfloor = \lceil f\rfloor = f$ (from Definition 41), and hence, this triangle is commutative. Analogously, we show that $g = [f,g] \circ \mathrm{in}_B$.

Let us suppose that there is another arrow $k' = [k_1,k_2] : A + B \to C$ such that the diagram above commutes, i. e., $k' \circ \mathrm{in}_A = f$ and $k' \circ \mathrm{in}_B = g$. However, $k' \circ \mathrm{in}_A = \lceil k_1, \perp^1\rfloor = k_1$ and $k' \circ \mathrm{in}_B = \lceil \perp^1, k_2\rfloor = k_2$, so that $k_1 = f$ and $k_2 = g$, i. e., $k' = k = [f,g]$ is a unique arrow that satisfies this commutativity. □

The product diagram is its dual with $k = [f,g]^{\mathrm{OP}} = \langle f^{\mathrm{OP}}, g^{\mathrm{OP}}\rangle$ (from Corollary 23):

$\times \equiv \biguplus$,



Let us show that the complex morphisms with coproducts satisfy the powerview functor T, the duality and the symmetry property in **DB**.

Corollary 23. *The complex morphisms satisfy the following* **DB** *properties:*

1. *(Power-view functorial properties)*

$$T(f+g) = Tf + Tg, \quad T([f,g]) = [Tf, Tg], \quad T(\langle f,g\rangle) = \langle Tf, Tg\rangle \quad \textit{and}$$
$$T(\lceil f,g\rfloor) = \lceil Tf, Tg\rfloor.$$

2. *(The duality properties)*

$$(f+g)^{\mathrm{OP}} = f^{\mathrm{OP}} + g^{\mathrm{OP}}, \quad [f,g]^{\mathrm{OP}} = \langle f^{\mathrm{OP}}, g^{\mathrm{OP}}\rangle, \quad \langle f,g\rangle^{\mathrm{OP}} = [f^{\mathrm{OP}}, g^{\mathrm{OP}}] \quad \textit{and}$$
$$\lceil f,g\rfloor^{\mathrm{OP}} = \lceil f^{\mathrm{OP}}, g^{\mathrm{OP}}\rfloor.$$

3. *(The symmetry property of Theorem* 14 *with* $\tau^{-1} \bullet \tau = \psi$*)*

$$\tau(J(f+g)) = \tau_{J(f)} + \tau_{J(g)}, \quad \tau^{-1}(J(f+g)) = \tau^{-1}_{J(f)} + \tau^{-1}_{J(g)},$$
$$\tau(J([f,g])) = \tau_{J(f)} + \tau_{J(g)}, \quad \tau^{-1}(J([f,g])) = [\tau^{-1}_{J(f)}, \tau^{-1}_{J(g)}],$$
$$\tau(J(\langle f,g\rangle)) = \langle \tau_{J(f)}, \tau_{J(g)}\rangle, \quad \tau^{-1}(J(\langle f,g\rangle)) = \tau^{-1}_{J(f)} + \tau^{-1}_{J(g)},$$
$$\tau(J(\lceil f,g\rfloor)) = \tau_{J(f)} \bigcup \tau_{J(g)}, \quad \tau^{-1}(J(\lceil f,g\rfloor)) = \tau^{-1}_{J(f)} \bigcup \tau^{-1}_{J(g)}.$$

Proof. Claim 1 (Power-view properties):

1.1 For $f+g : A+B \to C+D$, from Theorem 11, $T(f+g) = \mathrm{is}^{-1}_{C+D} \circ (f+g) \circ \mathrm{is}_{A+B} = (\mathrm{is}^{-1}_C + \mathrm{is}^{-1}_D) \circ (f+g) \circ (\mathrm{is}_A + \mathrm{is}_B) = (\mathrm{is}^{-1}_C \circ f \circ \mathrm{is}_A) + (\mathrm{is}^{-1}_D \circ g \circ \mathrm{is}_B) = Tf + Tg$.

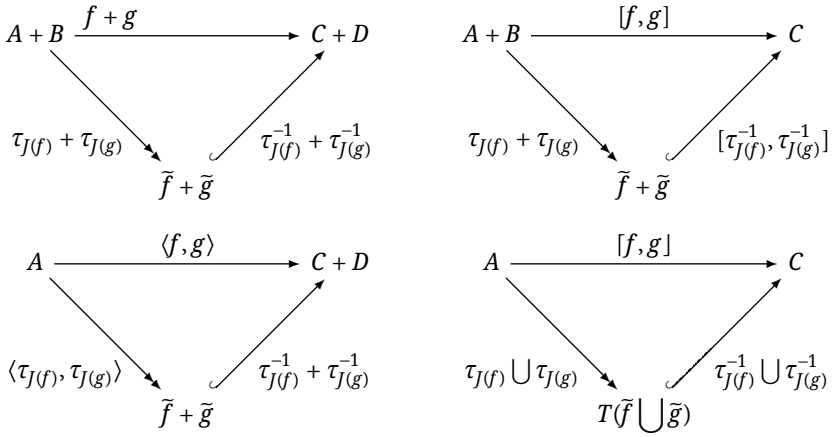
1.2 For $[f,g] : A+B \to C$, from Theorem 11, $T([f,g]) = \mathrm{is}_C \circ [f,g] \circ \mathrm{is}^{-1}_{A+B} = \mathrm{is}_C \circ [f,g] \circ (\mathrm{is}^{-1}_A + \mathrm{is}^{-1}_B) = [\mathrm{is}_C \circ f \circ \mathrm{is}^{-1}_A, \mathrm{is}_C \circ g \circ \mathrm{is}^{-1}_B] = [Tf, Tg]$.

The other two cases can be proved analogously.

Claim 2 (Duality). It holds from the fact that $\widetilde{f^{\mathrm{OP}}} = \widetilde{f}$. Let us show that 12 is applicable. For example, consider $[f,g] : A+B \to C$:

$$\begin{aligned}
[f,g]^{\mathrm{OP}} &= \left[\mathrm{is}^{-1}_C \circ Tf \circ \mathrm{is}_A, \mathrm{is}^{-1}_C \circ Tg \circ \mathrm{is}_B\right]^{\mathrm{OP}} \quad \text{from Theorem 11} \\
&= \left\langle \left(\mathrm{is}^{-1}_C \circ Tf \circ \mathrm{is}_A\right)^{\mathrm{OP}}, \left(\mathrm{is}^{-1}_C \circ Tg \circ \mathrm{is}_B\right)^{\mathrm{OP}} \right\rangle \\
&= \left\langle \mathrm{is}^{-1}_A \circ (Tf)^{\mathrm{OP}} \circ \mathrm{is}_C, \mathrm{is}^{-1}_B \circ (Tg)^{\mathrm{OP}} \circ \mathrm{is}_C \right\rangle \\
&= \left\langle f^{\mathrm{OP}}, g^{\mathrm{OP}} \right\rangle \quad \text{from Theorem 12.}
\end{aligned}$$

Claim 3. The extended symmetry $\tau^{-1} \bullet \tau = \psi$ can be shown by the following four commutative diagrams:



based on the results in point 1 and the facts of Theorem 14, i. e., $\widetilde{\tau_{J(f)}} = \widetilde{\tau^{\mathrm{OP}}_{J(f)}} = \widetilde{f}$.

For example, in the second case we obtain

$$\begin{aligned}[\tau^{-1}_{J(f)}, \tau^{-1}_{J(g)}] \circ (\tau_{J(f)} + \tau_{J(g)}) &= [\tau^{-1}_{J(f)}, \tau^{-1}_{J(g)}] \circ \left(\tau_{J(f)} \biguplus \tau_{J(g)}\right) \\ &= \text{(by rule 3 for composition of complex arrows)} \\ &= [\tau^{-1}_{J(f)} \circ \tau_{J(f)}, \tau^{-1}_{J(g)} \circ \tau_{J(g)}] = [f, g]. \end{aligned}$$ □

Let us show some examples for the symmetry-closure functor T_e of Theorem 14, applied to complex morphisms and (co)products.

Example 26. Let us consider a number of "conceptualizations" of complex morphisms in Theorem 14, expressed by the functor $T_e = (T^0_e, T^1_e) : (\mathbf{DB} \downarrow \mathbf{DB}) \longrightarrow \mathbf{DB}$:

1. The case when $f = f_1 + f_2 : A_1 + A_2 \to B_1 + B_2$, $g = g_1 + g_2 : C_1 + C_2 \to D_1 + D_2$, $h = h_1 + h_2 : A_1 + A_2 \to C_1 + C_2$ and $k = k_1 + k_2 : B_1 + B_2 \to D_1 + D_2$, such that $k_1 \circ f_1 = g_1 \circ h_1$ and $k_2 \circ f_2 = g_2 \circ h_2$, so that we have the arrow $(h_1 + h_2; k_1 + k_2) : J(f) \to J(g)$, and hence, $T_e(h_1 + h_2; k_1 + k_2) = T_e(h_1, k_1) + T_e(h_2, k_2) : \widetilde{f_1} + \widetilde{f_2} \to \widetilde{g_1} + \widetilde{g_2}$.
2. The case when $f = [f_1, f_2] : A_1 + A_2 \to B$, $g : C \to D$, $h = [h_1, h_2] : A_1 + A_2 \to C$ and $k : B \to D$, such that $k \circ f_1 = g \circ h_1$ and $k \circ f_2 = g \circ h_2$, so that we have the arrow $([h_1, h_2]; k) : J(f) \to J(g)$, and hence, $T_e([h_1, h_2]; k) = [T_e(h_1, k), T_e(h_2, k)] : \widetilde{f_1} + \widetilde{f_2} \to \widetilde{g}$.
3. The case when $f : A \to B$, $g = [g_1, g_2] : C_1 + C_2 \to D$, $h = \langle h_1, h_2 \rangle : A \to C_1 + C_2$ and $k : B \to D$, such that $k \circ f = g_1 \circ h_1$ and $k \circ f = g_2 \circ h_2$, so that we have the arrow $(\langle h_1, h_2 \rangle; k) : J(f) \to J(g)$, and hence, $T_e(\langle h_1, h_2 \rangle; k) = \langle T_e(h_1, k), T_e(h_2, k) \rangle : \widetilde{f} \to \widetilde{g_1} + \widetilde{g_2}$.
4. The case when $f = \langle f_1, f_2 \rangle : A \to B_1 + B_2$, $g = \langle g_1, g_2 \rangle : C \to D_1 + D_2$, $h : A \to C$ and $k = k_1 + k_2 : B_1 + B_2 \to D_1 + D_2$, such that $k_1 \circ f_1 = g_1 \circ h$ and $k_2 \circ f_2 = g_2 \circ h$, so that we have the arrow $(h; k_1 + k_2) : J(f) \to J(g)$, and hence, $T_e(h; k_1 + k_2) = \langle T_e(h, k_1), T_e(h, k_2) \rangle : \widetilde{f} \to \widetilde{g_1} + \widetilde{g_2}$.

5. The case when $f = \langle f_1, f_2\rangle : A \to B_1 + B_2$, $g : C \to D$, $h = \lceil h_1, h_2\rfloor : A \to C$ and $k = [k_1, k_2] : B_1 + B_2 \to D$, such that $k_1 \circ f_1 = g \circ h_1$ and $k_2 \circ f_2 = g \circ h_2$, so that we have the arrow $(\lceil h_1, h_2\rfloor; [k_1, k_2]) : J(f) \to J(g)$, and hence, $T_e(\lceil h_1, h_2\rfloor; [k_1, k_2]) = [T_e(h_1, k_1), T_e(h_2, k_2)] : \widetilde{f_1} + \widetilde{f_2} \to \widetilde{g}$.
   But also more complex cases:
6. The case when $f = [f_1, f_2] : A_1 + A_2 \to B$, $g = [g_1, g_2] : C_1 + C_2 \to D$, $h = h_1 + h_2 : A_1 + A_2 \to C_1 + C_2$ and $k : B \to D$, such that $k \circ f_1 = g_1 \circ h_1$ and $k \circ f_2 = g_2 \circ h_2$, so that we have the arrow $(h_1 + h_2; k) : J(f) \to J(g)$, and hence, $T_e(h_1 + h_2; k) = \langle \mathrm{in}_{g_1}, \mathrm{in}_{g_2}\rangle \circ T(k \circ [f_1, f_2]) \circ (\mathrm{in}_{f_1} + \mathrm{in}_{f_2}) = [\langle T_e(h_1, k), \mathrm{in}_{g_2} \circ T(k \circ f_1) \circ \mathrm{in}_{f_1}\rangle, \langle \mathrm{in}_{g_1} \circ T(k \circ f_2) \circ \mathrm{in}_{f_2}, T_e(h_2, k)\rangle] : \widetilde{f_1} + \widetilde{f_2} \to \widetilde{g_1} + \widetilde{g_2}$.
7. The case when $f = \langle f_1, f_2\rangle : A_1 + A_2 \to B$, $g = [g_1, g_2] : C_1 + C_2 \to D$, $h = \langle h_1, h_2\rangle : A_1 + A_2 \to C_1 + C_2$ and $k = [k_1, k_2] : B_1 + B_2 \to D_1 + D_2$, we obtain that $T_e(h_1 + h_2; [k_1, k_2]) = \lceil \langle [T_e(h_1, k_1), \mathrm{in}_{g_1} \circ T(k_1 \circ f_1) \circ \mathrm{in}_{f_2}], [\mathrm{in}_{g_2} \circ T(k_1 \circ f_1) \circ \mathrm{in}_{f_1}, \mathrm{in}_{g_2} \circ T(k_1 \circ f_1) \circ \mathrm{in}_{f_2}]\rangle, \langle [\mathrm{in}_{g_1} \circ T(k_2 \circ f_2) \circ \mathrm{in}_{f_1}, \mathrm{in}_{g_1} \circ T(k_2 \circ f_2) \circ \mathrm{in}_{f_2}], [\mathrm{in}_{g_2} \circ T(k_2 \circ f_2) \circ \mathrm{in}_{f_1}, T_e(h_2, k_2)]\rangle \rfloor : \widetilde{f_1} + \widetilde{f_2} \to \widetilde{g_1} + \widetilde{g_2}$.

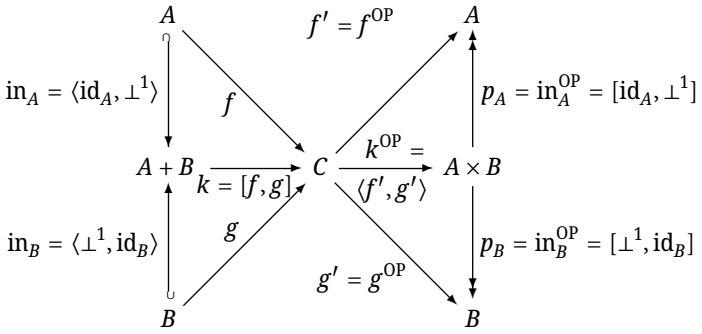
7.3.2 (Co)Limits and exponentiation

In previous section, we introduced the (co)products in the **DB** category that are operators used in order to make the separation composition of the databases. Here, instead, we will investigate the rest of the limits and colimits in **DB** category in order to establish the general properties of this category, used as a base category for the functorial semantics of schema database mapping systems.

In order to explain these concepts in another way, we can see the limits and colimits as a left and a right adjunction for the diagonal functor $\triangle : \mathbf{DB} \longrightarrow \mathbf{DB}^{\mathbf{J}}$ for a small index category (i. e., a diagram) **J**. For any (co)limit functor G, a right adjoint to the diagonal functor (i. e., the adjunction, $(\triangle, G, \varepsilon, \eta) : \mathbf{DB} \longrightarrow \mathbf{DB}^{\mathbf{J}}$) with the (co)limit object $G(D)$ and the universal (counit) cone (a natural transformation) $\varepsilon : \triangle G \xrightarrow{\cdot} \mathrm{Id}_{DB^{\mathbf{J}}}$, and $\eta : \mathrm{Id}_{\mathrm{DB}} \xrightarrow{\cdot} G\triangle$ (for colimits, unit and counit are inverted w. r. t. limits).

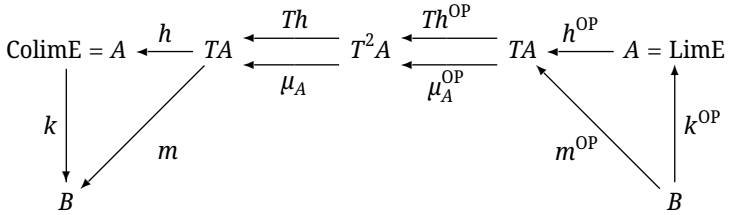
Let us consider, e. g., the coproducts ($G = + \equiv \biguplus$) and the products ($G = \times \equiv +$). In this case, the diagram (a functor) $D = (A, B) \in \mathbf{DB}^{\mathbf{J}}$ is just a diagram of two objects in **DB** without arrows between them (**J** is a discrete category with two index objects a_1 and a_2 and with their identity arrows, so that $\mathbf{DB}^{\mathbf{J}} \simeq \mathbf{DB}^2 = \mathbf{DB} \times \mathbf{DB}$).

We obtain for the universal cocone counit $\varepsilon(A, B) : (A, B) \longrightarrow (A + B, A + B)$, one pair of the coproduct inclusion-monomorphisms $\varepsilon(A, B) = (\mathrm{in}_A, \mathrm{in}_B)$ where $\mathrm{in}_A = \langle \mathrm{id}_A, \perp^1\rangle : A \hookrightarrow A + B$ and $\mathrm{in}_B = \langle \perp^1, \mathrm{id}_B\rangle : B \hookrightarrow A + B$. Dually, for the products, the universal cone counit of the product $\varepsilon(A, B) : (A \times B, A \times B) \longrightarrow (A, B)$ is a pair of the product projection-epimorphisms $\varepsilon(A, B) = (p_A, p_B)$ where $p_A : A \times B \twoheadrightarrow A$ and $p_B : A \times B \twoheadrightarrow B$. Moreover, $A \times B = A + B$, $p_A = \mathrm{in}_A^{\mathrm{OP}} = [\mathrm{id}_A, \perp^1]$ and $p_B = \mathrm{in}_B^{\mathrm{OP}} = [\perp^1, \mathrm{id}_B]$, as represented in the following diagram:



In the case when **J** is an index category with two objects and two parallel arrows from the first into the second object, each diagram (a functor) $D \in \mathbf{DB}^{\mathbf{J}}$ is of the form $B \overset{f}{\underset{g}{\leftarrow\!\leftarrow}} A$. The limits of such diagrams are the equalizers, while their colimits are the coequalizers, as in the following example.

Example 27. Let us verify that each object in **DB** is a limit of some equalizer and a colimit of its dual coequalizer. In fact, for any simple object A, a "structure map" $h : TA \longrightarrow A$ of a *monadic* T-algebra (A, h) ($h : TA \to A$ is an isomorphism, thus epic and monic as well) derived from a monad (T, η, μ), where from Definition 46, in Section 7.1.2, $h \circ \eta_A = \mathrm{id}_A$, and $h \circ Th = h \circ \mu_A$, we obtain the *coequalizer* (by Back's theorem, it is preserved by the endofunctor T, i. e., T creates a coequalizer) with a colimit A, and, by duality, we obtain the absolute equalizer with the limit A as well:



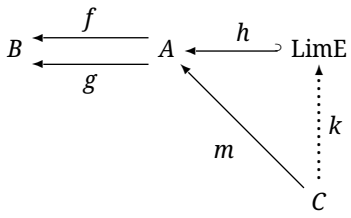
That is, whenever $m : TA \to B$ (with a simple object B) has $m \circ Th = m \circ \mu_A$ in **DB**, there is exactly one arrow $k : A \to B$ such that $m = k \circ h$. In fact, if $m \circ Th = m \circ \mu_A$ then $\widetilde{m} \bigcap \widetilde{Th} = \widetilde{m} \bigcap \widetilde{\mu_A}$, and from $\widetilde{m} \subseteq TA = \widetilde{\mu_A}$ and $\widetilde{Th} = \widetilde{h}$ we obtain $\widetilde{m} \bigcap \widetilde{h} = \widetilde{m}$, i. e., $\widetilde{m} \subseteq \widetilde{h}$.

Consequently, in order to satisfy the commutativity $m = k \circ h$, i. e., $\widetilde{m} = \widetilde{k} \bigcap \widetilde{h}$, we must have that $\widetilde{k} = \widetilde{m}$, so that k is uniquely determined by the morphism m.

Analogously, we obtain for the equalizer in the right part of the diagram above.

Lemma 8. **DB** *has the (co)equalizers for every pair of morphisms.*

Proof. For each pair of simple morphisms $f, g : A \to B$, and hence, $\widetilde{f} \subseteq TA$ and $\widetilde{g} \subseteq TA$, the $\mathrm{LimE} = (\widetilde{f} \bigcap \widetilde{g}) \bigcup (TA \backslash (\widetilde{f} \bigcup \widetilde{g})) \subseteq TA$ is a limit object (up to isomorphism) of the diagram $B \overset{f}{\underset{g}{\leftarrow\!\leftarrow}} A$, so that the following diagram is an equalizer:



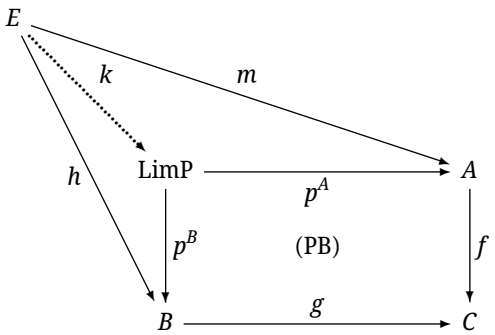
with
(1) $f \circ h = g \circ h$,
(2) $f \circ m = g \circ m$,

where h is a monomorphism (every equalizer is monic). It is easy to verify that $\widetilde{h} = T(\text{LimE}) \subseteq TA$ is the maximal flux such that $\widetilde{f} \bigcap \widetilde{h} = \widetilde{g} \bigcap \widetilde{h}$ (equal to $\widetilde{f} \bigcap \widetilde{g}$). From (2), we have $\widetilde{f} \bigcap \widetilde{m} = \widetilde{g} \bigcap \widetilde{m}$, so that it must be $\widetilde{m} \subseteq \widetilde{h}$, and hence, for each $m : C \to A$ such that (2) is true, there is a unique $k : C \to \text{LimE}$ with $\widetilde{k} = \widetilde{m}$ for which $m = h \circ k$. By duality, it is valid for coequalizers as well. The equalizers for the complex arrows are given as a particular case of the pullbacks after the proof of next Proposition 31. □

In the case when **J** is an index category with three objects and two arrows with the same codomain, each diagram (a functor) $D \in \mathbf{DB}^{\mathbf{J}}$ is of the form $A \xrightarrow{f} C \xleftarrow{g} B$. The limits of such diagrams are the pullbacks. In the case when **J** is an index category with one object only, the limit of such diagram corresponds to a terminal object of such a category. A category is complete if every diagram in it has a limit.

Dually, it is cocomplete when every diagram has a colimit. A *finite* diagram (a functor in $\mathbf{DB}^{\mathbf{J}}$ in our case) is one that has a finite number of objects, and a finite number of arrows between them. A category is *finitely complete* (Cartesian) if it has a limit for every finite diagram. It is well known that any category that has a terminal object and the pullbacks for every pair of morphisms, then it has *all* finite limits.

Let us show that **DB** is not finitely complete if we reduce only to simple arrows and objects. For example, for the pullbacks. Let us suppose that the following diagram, where A, B and C are simple databases and LimP is a limit object of the diagram $A \xrightarrow{f} C \xleftarrow{g} B$, is a pullback (PB):



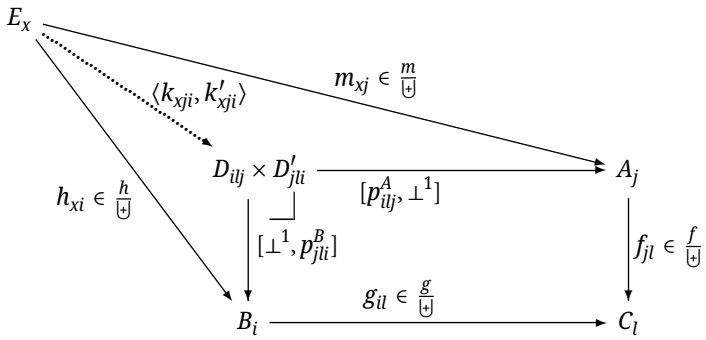
In order to have for every pair of arrows (m,h), the commutativity $f\circ m = g\circ h$, i. e., $\widetilde{f}\bigcap\widetilde{m} = \widetilde{g}\bigcap\widetilde{h}$ it must be true that $\widetilde{m}\subseteq T((TA\bigcap\widetilde{g})\bigcup(TA\backslash\widetilde{f}))$ and $\widetilde{h}\subseteq T((TB\bigcap\widetilde{f})\bigcup(TB\backslash\widetilde{g}))$, and from the commutativity $m = p^A\circ k$ and $h = p^B\circ k$, $\widetilde{m}\subseteq\widetilde{p^A}$ and $\widetilde{h}\subseteq\widetilde{p^B}$, it must be true that $\widetilde{p^A} = T((TA\bigcap\widetilde{g})\bigcup(TA\backslash\widetilde{f}))$ and $\widetilde{p^B} = T((TB\bigcap\widetilde{f})\bigcup(TB\backslash\widetilde{g}))$. In fact, from $\widetilde{f}\subseteq TA$ we have that $((TA\bigcap\widetilde{g})\bigcup(TA\backslash\widetilde{f}))\bigcap\widetilde{f} = \widetilde{f}\bigcap\widetilde{g}$, and hence, for closed object $\widetilde{f}\bigcap\widetilde{g} = T(\widetilde{f}\bigcap\widetilde{g}) = T(((TA\bigcap\widetilde{g})\bigcup(TA\backslash\widetilde{f}))\bigcap\widetilde{f}) = T((TA\bigcap\widetilde{g})\bigcup(TA\backslash\widetilde{f}))\bigcap\widetilde{f} = \widetilde{p^A}\bigcap\widetilde{f}$. Similarly, $\widetilde{p^B}\bigcap\widetilde{g} = \widetilde{f}\bigcap\widetilde{g}$, and hence, $\widetilde{p^A}\bigcap\widetilde{f} = \widetilde{p^B}\bigcap\widetilde{g}$, i. e., $f\circ p^A = g\circ p^B$.

However, if $\widetilde{m}\neq\widetilde{h}$ then there is no simple arrow $k : D\to$ LimP because we need $\widetilde{k} = \widetilde{h}$ (from $h = p^A\circ k$ and $\widetilde{h}\subseteq\widetilde{p^A}$), and similarly $\widetilde{k} = \widetilde{m}$, and hence, $\widetilde{m} = \widetilde{h}$, which is a contradiction (the case when $\widetilde{m} = \widetilde{h}$ with $A = B$ is the case when it is a pullback with all simple objects, corresponding to equalizers in Lemma 8). Consequently, LimP $= ((TA\bigcap\widetilde{g})\bigcup(TA\backslash\widetilde{f}))\times((TB\bigcap\widetilde{f})\bigcup(TB\backslash\widetilde{g}))$ and all arrows to and from it have to be *complex*, i. e., $k = \langle m,h\rangle$ and p^B and p^A substituted by $[\bot^1,p^B]$ and $[p^A,\bot^1]$, respectively. Consequently, we obtain the following.

Proposition 31. **DB** *category is* finitely (co)complete*, i. e., (co)Cartesian.*

Proof. Let us consider a pullback diagram (PB) above with $A = \biguplus_{1\le j\le M}A_j$, $B = \biguplus_{1\le i\le K}B_i$, $C = \biguplus_{1\le l\le N}C_l$ and $E = \biguplus_{1\le x\le N_1}E_x$ for $M,K,N,N_1\ge 1$. Then a limit cone of this pullback is composed by limit object (see the ptp arrows in S_A and S_B in the diagram below) LimP $= S_E\biguplus S_A\biguplus S_B$ where $S_E = \{D_{ilj} = D'_{jli} = (\widetilde{f_{jl}}\bigcap\widetilde{g_{il}})\bigcup(TA_j\backslash(\widetilde{f_{jl}}\bigcup\widetilde{g_{il}}) \mid f_{jl}\in\frac{f}{\biguplus},g_{il}\in\frac{g}{\biguplus},i=j$ and $TB_i = TA_j\}$, $S_A = \{D_{ilj} = (TA_j\bigcap\widetilde{g_{il}})\bigcup(TA_j\backslash\widetilde{f_{jl}}) \mid f_{jl}\in\frac{f}{\biguplus},g_{il}\in\frac{g}{\biguplus},TB_i\neq TA_j\}$ and $S_B = \{D'_{jli} = (TB_i\bigcap\widetilde{f_{jl}})\bigcup(TB_i\backslash\widetilde{g_{il}}) \mid f_{jl}\in\frac{f}{\biguplus},g_{il}\in\frac{g}{\biguplus},TB_i\neq TA_j\}$, and with outgoing arrows $\frac{p^A}{\biguplus} = \{p^A_{ilj} : D_{ilj}\hookrightarrow A_j \mid D_{ilj}\in S_E\bigcup S_A\}$ and $\frac{p^B}{\biguplus} = \{p^B_{jli} : D'_{jli}\hookrightarrow B_i \mid D'_{jli}\in S_E\bigcup S_B\}$. From the fact that $\frac{f\circ p^A}{\biguplus} = \{f_{jl}\circ p^A_{ijl} : D_{ilj}\to C_l \mid f_{jl}\in\frac{f}{\biguplus},g_{il}\in\frac{g}{\biguplus}\}$ and $\frac{g\circ p^B}{\biguplus} = \{g_{il}\circ p^B_{jli} \mid f_{jl}\in\frac{f}{\biguplus},g_{il}\in\frac{g}{\biguplus}\}$, we obtain the bijection $\sigma : \frac{f\circ p^A}{\biguplus}\to\frac{g\circ p^B}{\biguplus}$, such that for each $f_{ji}\circ p^A_{ijl}\in\frac{f}{\biguplus}$, $\sigma(f_{ji}\circ p^A_{ijl}) = g_{il}\circ p^B_{jli}$ with $\widetilde{\sigma(f_{ji}\circ p^A_{ijl})} = \widetilde{g_{il}}\bigcap\widetilde{p^B_{jli}} = \widetilde{g_{il}}\bigcap\widetilde{f_{jl}} = \widetilde{p^A_{ijl}}\bigcap\widetilde{f_{jl}} = \widetilde{f_{ji}\circ p^A_{ijl}}$ and, consequently we have the same set of ptp arrows $\frac{f\circ p^A}{\biguplus} = \frac{g\circ p^B}{\biguplus}$, i. e., the commutativity of the square diagram (PB), i. e., $f\circ p^A = g\circ p^B$.

Suppose that for another two arrows $m : E\to A$ and $h : E\to B$ the diagram (PB) commutes, i. e., $f\circ m = g\circ h$. For $\frac{m}{\biguplus} = \frac{h}{\biguplus} = \emptyset$, it is obvious, so we will consider nontrivial cases when both arrows have a nonempty set of ptp arrows. Let us define the sets (see the commutative diagram below composed by the ptp arrows of this pullback diagram), $S_1 = \{k_{xji} : E_x\to D_{ilj} \mid m_{xj}\in\frac{m}{\biguplus},p^A_{ilj}\in p^A$ with $\widetilde{k_{xji}} = \widetilde{m_{xj}}\subseteq\widetilde{p^A_{ilj}}\}$, and $S_2 = \{k'_{xji} : E_x\to D'_{jli} \mid h_{xi}\in\frac{h}{\biguplus},p^B_{jli}\in p^B$ with $\widetilde{k'_{xji}} = \widetilde{h_{xi}}\subseteq\widetilde{p^B_{jli}}\}$. Then we define the arrow $k : E\to$ LimP as the canonical representation (from Lemma 2) composed by the set of ptp arrows $\frac{k}{\biguplus}\triangleq S_1\bigcup S_2$:



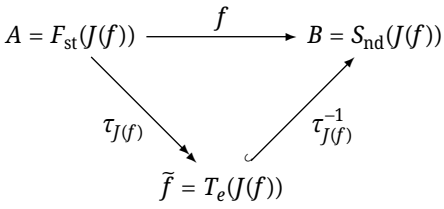
Let us show that the pullback diagram (PB) commutes for this arrow k, i. e., $p^B \circ k = h$ (and $p^A \circ k = m$). First of all, note that each ptp arrow $(s_{xi} : E_x \to B_i) \in \frac{p^B \circ k}{\biguplus}$ is the union of the following set of compositions of the ptp arrows in $S_2 \subset \frac{k}{\biguplus}$ and in $\frac{p^B}{\biguplus}$, $s_{xi} = \bigcup\{p^B_{jli} \circ k'_{xji} : E_x \to B_i \mid k'_{xji} \in S_2, p^B_{jli} \in p^B \text{ and } 1 \leq j \leq M\}$.

From Lemma 1, the information flux of this ptp arrow is $\widetilde{s_{xi}} = T(\bigcup\{\widetilde{p^B_{jli} \circ k'_{xji}} \mid p^B_{jli} \circ k'_{xji} \in \frac{s_{xi}}{\biguplus}\}) = T(\bigcup\{\widetilde{p^B_{jli}} \bigcap \widetilde{k'_{xji}} \mid p^B_{jli} \circ k'_{xji} \in \frac{s_{xi}}{\biguplus}\}) = T(\bigcup\{\widetilde{p^B_{jli}} \bigcap \widetilde{h_{xi}} \mid p^B_{jli} \circ k'_{xji} \in \frac{s_{xi}}{\biguplus}\}) = T(\bigcup\{\widetilde{h_{xi}} \mid p^B_{jli} \circ k'_{xji} \in \frac{s_{xi}}{\biguplus}\}) = T(\widetilde{h_{xi}}) = \widetilde{h_{xi}}$, i. e., $s_{x_i} = h_{xi}$. Consequently, $\frac{p^B \circ k}{\biguplus} = \frac{h}{\biguplus}$, and hence, $p^B \circ k = h$. Analogously, $p^A \circ k = m$, so that k is this unique arrow, which satisfies the pullback's requirements. □

Notice that if this definition of the pullbacks we considered only nonempty arrows f and g. Notice that when $\frac{f}{\biguplus} = \frac{g}{\biguplus} = \emptyset$ are empty morphisms, then the pullback diagram corresponds to the standard product diagram. In this case, $D_{ilj} = (TA_j \bigcap \perp^0) \bigcup (TA_j \setminus \perp^0) = TA_j \simeq A_j$, $D'_{jli} = (TB_i \bigcap \perp^0) \bigcup (TB_i \setminus \perp^0) = TB_i \simeq B_i$. Thus, $\mathrm{LimP} = A \times B$ with the projections $p^A = [\mathrm{id}_A, \perp^1]$ and $p^B = [\perp^1, \mathrm{id}_B]$.

In fact, when $B = A$ and $S_A = S_B = \emptyset$, we obtain $p^A = p^B$ with limit object $\mathrm{LimP} = \biguplus S_E$ corresponding to the equalizer.

From the fact that **DB** is an *symmetry-extended* category, for any morphism $f : A \to B$ (i. e., each object $J(f) = \langle A, B, f\rangle$ in **DB** $\downarrow$ **DB**), the following diagram commutes (see the proof of Theorem 14 in Section 7.3):



such that

$$T_e(J(f)) = \widetilde{f} \simeq \widetilde{\biguplus_{\biguplus} \frac{f}{\biguplus}} = \biguplus_{f_{ji} \in \frac{f}{\biguplus}} \widetilde{f_{ji}}, \quad \text{and}$$

$$\frac{\tau_{J(f)}}{\biguplus} = \left\{ \tau_{J(f_{ji})} : A_j \twoheadrightarrow \widetilde{f_{ji}} \mid (f_{ji} : A_j \to B_i) \in \frac{f}{\biguplus} \right\}, \quad \text{and}$$

$$\frac{\tau^{-1}_{J(f)}}{\biguplus} = \left\{ \tau^{-1}_{J(f_{ji})} : \widetilde{f_{ji}} \hookrightarrow B_i \mid (f_{ji} : A_j \to B_i) \in \frac{f}{\biguplus} \right\}.$$

This decomposition of each morphism f into one epic $\tau_{J(f)} : A \twoheadrightarrow \widetilde{f}$ and one monic $\tau^{-1}_{J(f)} : \widetilde{f} \hookrightarrow B$ arrows will be used to demonstrate the existence of a limit in **DB** for each diagram D composed of a single morphism $f : A \to B$ in **DB**, by the pullback of the diagram $B \xrightarrow{\mathrm{id}_B} B \xleftarrow{f} A$ (when $C = B$ and $g = \mathrm{id}_B$). Here, we will consider the following particular cases for f.

Corollary 24. *For any given morphism $f : A \to B$ such that for each its ptp arrow $f_{ji} : A_j \to B_i$, $TA_j \subseteq TB_i$, with $A = \biguplus_{1 \le j \le M} A_j$, $M \ge 1$ and $B = \biguplus_{1 \le j \le K} B_i$, $K \ge 1$, and a diagram $B \xrightarrow{\mathrm{id}_B} B \xleftarrow{f} A$, the cone with the vertex*

$$\mathrm{Lim}(f) = \biguplus_{(f_{ji} : A_j \to B_i) \in \frac{f}{\biguplus}} (A_j \times \widetilde{f_{ji}}),$$

and with outgoing arrows $\frac{p^B}{\biguplus} = \{\tau^{-1}_{J(f_{ji})} : \widetilde{f_{ji}} \hookrightarrow B_i \mid f_{ji} \in \frac{f}{\biguplus}\} = \frac{\tau^{-1}_{J(f)}}{\biguplus}$ and $\frac{p^A}{\biguplus} = \frac{\mathrm{id}_A}{\biguplus}$, defines a pullback diagram,

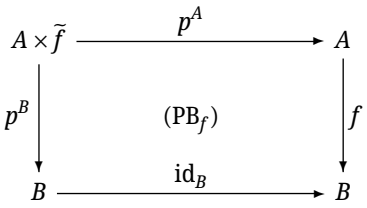
$$\begin{array}{ccc} \mathrm{Lim}(f) & \xrightarrow{\quad p^A \quad} & A \\ \Big\downarrow{\scriptstyle p^B} & (\mathrm{PB}_f) & \Big\downarrow{\scriptstyle f} \\ B & \xrightarrow{\quad \mathrm{id}_B \quad} & B \end{array}$$

This diagram can be simplified by substituting the limit object $\mathrm{Lim}(f)$ *with the object $A \times \widetilde{f}$ isomorphic to it.*

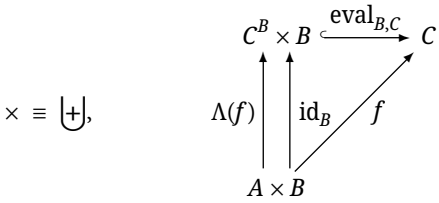
Proof. For a morphism f, we have that for each of its ptp arrows $f_{ji} : A_j \to B_i$, $\widetilde{f_{ji}} \subseteq TA_j \subseteq TB_i$. The first part comes directly from the proof of Proposition 31 in the case when $C = B$ and $g = \mathrm{id}_B$, with $l = i$, $D_{ilj} = (TA_j \bigcap \widetilde{g_{ii}}) \bigcup (TA_j \backslash \widetilde{f_{ji}}) = (TA_j \bigcap TB_i) \bigcup (TA_j \backslash \widetilde{f_{ji}}) = TA_j \simeq A_j$ and $D'_{jli} = (TB_i \bigcap \widetilde{f_{ji}}) \bigcup (TB_i \backslash \widetilde{g_{ii}}) = \widetilde{f_{ji}} \bigcup (TB_i \backslash TB_i) = \widetilde{f_{ji}}$.

From the fact that f is a complex monomorphism, we have a ptp arrow for every simple object A_j, $1 \le j \le M$, so from the commutativity we have that $\mathrm{Lim}(f) \simeq A_1 \biguplus \cdots \biguplus A_M \biguplus \frac{\widetilde{f}}{\biguplus} \simeq A \times \widetilde{f}$ (we recall that the product '$\times$' is equal to $\biguplus$). The rest comes directly from it. □

Remark. Consequently, for each morphism $f : A \to B$ such that for each of its ptp arrows $f_{ji} : A_j \to B_i$, $TA_j \subseteq TB_i$, the following commutative diagram with the arrows p^B and p^A such that $\frac{p^B}{\cup} = \frac{\tau^{-1}_{J(f)}}{\cup}$ and $\frac{p^A}{\cup} = \frac{\mathrm{id}_A}{\cup}$ is a pullback diagram:



If the category is finitely complete and has the *exponentiation* as well then it is a Cartesian Closed Category (CCC). We say that a category has exponentiation if it has a product for any two objects, and if for any given objects B and C there is an object C^B and an arrow $\mathrm{eval}_{B,C} : C^B \times B \to C$, called an *evolution* arrow, such that for any object A and arrow $f : A \times B \to C$, there is a unique arrow $\Lambda(f) : A \to C^B$ making the following diagram commute:



such that $\mathrm{eval}_{B,C} \circ (\Lambda(f) \times \mathrm{id}_B) = f$. The assignment of $\Lambda(f)$ to f establishes a bijection Λ from the arrows from $A \times B$ to C and the arrows from A to C^B.

It is well known that in a CCC with a zero object (i. e., when terminal and initial objects are isomorphic) such a category degenerates, i. e., *all* objects in such a category are isomorphic. The **DB** category has a zero object $\perp^0$ but does not degenerate: this means that it cannot be a CCC. In what follows, we will demonstrate this fact by demonstrating that it has no exponentiation.

Proposition 32. **DB** *is not a CCC.*

Proof. Let C^B be the exponent object of the hom-set $\mathbf{DB}(C, B)$, which as a set of the morphisms between C and D, cannot be an object of **DB** (differently from **Set** category). Hence, we have to construct this object $C^B \in \mathbf{DB}$ (to "internalize" the hom-set $\mathbf{DB}(C, B)$). For any two simple objects $B = \alpha^*(\mathcal{B})$ and $C = \alpha^*(\mathcal{C})$, the set of all arrows $\{f_1, f_2, \ldots\} : B \to C$ where each simple arrow is $f_i = \alpha^*(\mathbf{M}^{(i)}_{BC})$, can be represented by a unique arrow $g = (\bigcup_{f_i \in \mathbf{DB}(B,C)} f_i) : B \to C$ (note that g is a unique ptp arrow of the complex arrow $\lceil f_1, f_2, \ldots \rfloor : B \to C$ representing the set of all arrows from B into C), so we define the object C^B to be the information flux (from Definition 67, Section A.3.5 in

the Appendix) of this arrow $g = \alpha^*(\bigcup\{\mathbf{M}^{(i)}_{BC} = \text{MakeOperads}(\mathcal{M}^{(i)}_{BC}) \mid \mathcal{M}^{(i)}_{BC} : \mathcal{B} \to \mathcal{C}\}) = \bigcup_{\mathcal{M}^{(i)}_{BC}:\mathcal{B}\to\mathcal{C}} \alpha^*(\mathbf{M}^{(i)}_{BC}) = \bigcup_{f_i\in\mathbf{DB}(B,C)} f_i$. From point 2 of Definition 67,

$$\Delta\Big(\alpha, \bigcup_{\mathcal{M}^{(i)}_{BC}:\mathcal{B}\to\mathcal{C}} \mathbf{M}^{(i)}_{BC}\Big) = \bigcup_{\mathcal{M}^{(i)}_{BC}:\mathcal{B}\to\mathcal{C}} \Delta(\alpha, \mathbf{M}^{(i)}_{BC}).$$

Consequently,

$$\begin{aligned}
\widetilde{g} &= \text{Flux}\Big(\alpha \bigcup_{\mathcal{M}^{(i)}_{BC}:\mathcal{B}\to\mathcal{C}} \mathbf{M}^{(i)}_{BC}\Big) = T\Big(\Delta\Big(\alpha, \bigcup_{\mathcal{M}^{(i)}_{BC}:\mathcal{B}\to\mathcal{C}} \mathbf{M}^{(i)}_{BC}\Big)\Big) \\
&= T\Big(\bigcup_{\mathcal{M}^{(i)}_{BC}:\mathcal{B}\to\mathcal{C}} \Delta(\alpha, \mathbf{M}^{(i)}_{BC})\Big) = T\Big(\bigcup_{\mathcal{M}^{(i)}_{BC}:\mathcal{B}\to\mathcal{C}} T(\Delta(\alpha, \mathbf{M}^{(i)}_{BC}))\Big) \\
&= T\Big(\bigcup_{\mathcal{M}^{(i)}_{BC}:\mathcal{B}\to\mathcal{C}} \text{Flux}(\alpha, \mathbf{M}^{(i)}_{BC})\Big) = T\Big(\bigcup_{f_i\in\mathbf{DB}(B,C)} \widetilde{f}_i\Big) = \oplus\{\widetilde{f}_i \mid f_i \in \mathbf{DB}(B,C)\}.
\end{aligned}$$

Thus, we define the hom-object $C^B \triangleq \oplus\{\widetilde{f}_i \mid f_i \in \mathbf{DB}(B,C)\}$, which "internalizes" the hom-sets (i. e., the *merging* of all closed objects obtained from a hom-set of arrows from B to C), i. e., the merging of all compact elements A such that $A \preceq B \otimes C$.

Hence, we obtain that $C^B = T(\bigcup\{\widetilde{f}_i \mid f_i \in \mathbf{DB}(B,C)\}) =$ (from Proposition 24) $= T(\bigcup\{\widetilde{f}_i \mid \widetilde{f}_i \subseteq B\otimes C\}) = T(\bigcup\{\widetilde{f}_i \mid \widetilde{f}_i \preceq B\otimes C\}) =$ (from algebraic lattice property, Section 7.4.4) $= T(B \otimes C) = B \otimes C$.

To have a Cartesian closed **DB** category (from Theorem 15 in Section 7.3.1, $\times \equiv \biguplus$), for any complex arrow $f = [f_1, f_2] : A \biguplus B \longrightarrow C$ in **DB**, we must have the exponential diagram commutativity, i. e., $f = \text{eval}_{B,C} \circ (\Lambda(f) \biguplus \text{id}_B)$, where $\Lambda(f) : A \longrightarrow C^B$ is a simple derived morphism by "abstraction" Λ and the complex arrow $\text{eval}_{B,C} = [e_1, e_2] : C^B \times B \longrightarrow C$ is a universal "application" morphism. Thus, from the commutativity in **DB** of such an exponential diagram, we obtain $f_2 = e_2 : B \to C$ and $f_1 = e_1 \circ \Lambda(f) : A \to C$, thus $\widetilde{f}_1 = \widetilde{e_1} \bigcap \widetilde{\Lambda(f)}$.

However, $\widetilde{f}_1 \subseteq TA \bigcap TC$ while $\widetilde{e_1} \bigcap \widetilde{\Lambda(f)} \subseteq (TA \bigcap TC) \bigcap TB$, so that there are the morphisms f_1 for which $\widetilde{f}_1 \subset \widetilde{e_1} \bigcap \widetilde{\Lambda(f)}$, and hence, the exponential diagram is not commutative. That is, **DB** has no exponentiation, and hence, is not a CCC. □

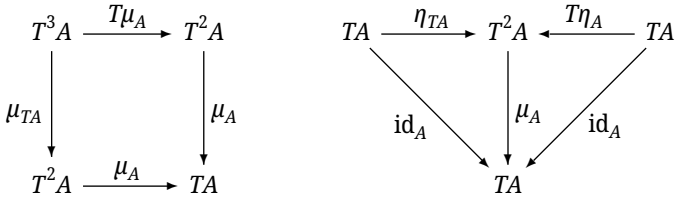
Remark. From the point 1 of Proposition 26 and the proof of this proposition, the "exponential" object C^B, which "internalizes" in **DB** the hom-set of its morphisms $\mathbf{DB}(B,C)$ is equal to the information flux of the *principal morphism* from B into C. □

Thus, from the fact that **DB** is not a CCC, we deduce that it is not a standard topos. But it still can be a kind of a *weak topos* [5], if we substitute the exponentiation with a kind of weak exponentiation (by replacing a (strong) product with a kind of (weak) tensorial product) and define a proper *subobject classifier* in **DB**.

7.3.3 Kleisli semantics for database mappings

We used monads to provide the *denotational semantics to database mappings*: in order to interpret database mappings (i. e., morphisms) in the category **DB**, we distinguish the object A (a database instance of type (or schema) $\mathcal{A}$) from the object TA of observations (i. e., the computations of type $\mathcal{A}$ without side-effects), and take as a denotation of (view) mappings the elements of TA (which are the views of (type) $\mathcal{A}$). In particular, we identify the A with the object of values (of type $\mathcal{A}$), and hence, we obtain the object of observations by applying the unary type-constructor T (power-view operator introduced in Section 7.1.2) to A.

We recall that a monad (T, η, μ) (from Proposition 27 in Section 7.1.2, the endofunctor $T : \mathbf{DB} \to \mathbf{DB}$ is a monad), given by commutative diagrams,



defines an adjunction $(F^T, G^T, \eta^T, \epsilon^T) : \mathbf{DB} \longrightarrow CT_{\mathrm{alg}}$ such that $G^T \circ F^T = T : \mathbf{DB} \longrightarrow \mathbf{DB}$, $\eta^T = \eta$, $\epsilon^T = \eta^{\mathrm{OP}}$ and $\mu = G^T \epsilon^T F^T$. The functors $F^T : \mathbf{DB} \to CT_{\mathrm{alg}}$ (into the Eilenberg–Moore category of *monadic* T-algebras, associated to the monad (T, η, μ)) and forgetful functor $G^T : CT_{\mathrm{alg}} \to \mathbf{DB}$ are defined as follows: for any object (database) A, $F^T(A) = (TA, \mu_A : T^2A \to TA)$, $F^T(f) = Tf$ while $G^T(A, g : TA \to A) = A$ and $G^T(f) = f$.

Let us consider the following properties for monadic algebras/coalgebras in **DB** (introduced by Definition 45 and 46 in Section 7.1.2).

Proposition 33. *The following properties for the monad (T, η, μ) and the comonad (T, η^C, μ^C) hold:*

- *The Eilenberg–Moore categories CT_{alg} and CT_{coalg} of the monad $T : \mathbf{DB} \longrightarrow \mathbf{DB}$ (from Proposition 27) are isomorphic ($CT_{\mathrm{coalg}} = CT_{\mathrm{alg}}^{\mathrm{OP}}$), complete and cocomplete. The object $(\perp^0, \mathrm{id}_{\perp^0} : \perp^0 \longrightarrow \perp^0)$ is an initial T-algebra in CT_{alg} and a terminal T-coalgebra in CT_{coalg}.*
- *For each object A in **DB** category there exist the unique monadic T-algebra $(A, \eta_A^C : TA \longrightarrow A)$ and the unique comonadic T-coalgebra $(A, \eta_A : A \longrightarrow TA)$ with $\eta_A^C = \eta_A^{\mathrm{OP}}$ (i. e., $\eta_A^C = \mathrm{is}_A^{-1} \approx \eta_A = \mathrm{is}_A \approx \mathrm{id}_A$).*
- *The free monadic T-algebra $(TA, \mu_A : T^2A \longrightarrow TA)$ is dual (and equal) to the cofree monadic T-coalgebra $(TA, \mu_A^C : TA \longrightarrow T^2A)$ with $\mu_A^C = \mu_A^{\mathrm{OP}}$ (i. e., $\mu_A^C = \mu_A = \mathrm{id}_{TA}$).*

Proof. Let us define the functor $F : CT_{\mathrm{alg}} \longrightarrow CT_{\mathrm{coalg}}$, such that for any T-algebra $(A, h : TA \longrightarrow A)$ we obtain the dual T-coalgebra $F^0(A, h) = (A, h^{\mathrm{OP}} : A \longrightarrow TA)$, where the component F^1 for the arrows is an identity function; and the functor $F : CT_{\mathrm{coalg}} \longrightarrow CT_{\mathrm{alg}}$,

such that for any T-coalgebra $(A, k : A \longrightarrow TA)$ we obtain the dual T-algebra $G^0(A,k) = (A, k^{\mathrm{OP}} : TA \longrightarrow A)$, where the component G^1 for the arrows is an identity function. Hence, $FG = I_{T_{\mathrm{coalg}}}$ and $GF = I_{T_{\mathrm{alg}}}$. The categories CT_{alg} and CT_{coalg} are complete and cocomplete as **DB** category ($CT_{\mathrm{coalg}} = CT_{\mathrm{alg}}^{\mathrm{OP}}$). The rest is easy to verify (by Proposition 27 and Corollary 20 in Section 7.1.2): each monadic T-algebra/coalgebra is an isomorphism. The free monadic T-algebra and the cofree monadic T-coalgebra are equal because $TA = T^2A$; thus, μ_A, μ_A^C are identity arrows (from duality property of **DB**). □

As we can see, each monadic T-coalgebra is an *equivalent inverted* arrow in **DB** of some monadic T-algebra and vice versa: the fundamental duality property of **DB** introduces the equivalence of monadic T-algebras and monadic T-coalgebras, and hence, the equivalence of the dichotomy "*construction* versus *observation*" or duality between induction and coinduction principles [57].

The basic idea behind the semantic of programs [95] is that a program denotes a morphism from A (the object of *values* of type A) to TB (the object of *computations* of type B), according to the view of "programs as functions from values to computations," so that the natural category for interpreting programs (in our case, a particular equivalent "computation" database mappings of the form $f_1 \triangleq \eta_B \circ f : A \longrightarrow TB$, derived from a database mapping $f : A \longrightarrow B$, such that $f_1 \approx f$) is not **DB** category, but the Kleisli category $\mathbf{DB}_T$.

Definition 50. Given a monad (T, η, μ) over a category **C**, we have [12]:

- A *Kleisli triple* is a triple $(T, \eta, -^*)$, where for $f : A \longrightarrow TB$ we have $f^* : TA \longrightarrow TB$ such that the following equations hold: $\eta_A^* = \mathrm{id}_{TA}, f^* \circ \eta_A = f$ and $g^* \circ f^* = (g^* \circ f)^*$, for $f : A \longrightarrow TB$ and $g : B \longrightarrow TC$.
  A Kleisli triple satisfies the *mono requirement* provided η_A is monic for each object A.
- A *Kleisli category* $\mathbf{C}_T$ has the same objects as **C** category. For any two objects A and B, there is the bijection between arrows $\theta : \mathbf{C}(A, TB) \cong \mathbf{C}_T(A, B)$. For any two arrows $f : A \longrightarrow B$ and $g : B \longrightarrow C$ in $\mathbf{C}_T$, their composition is defined by $g \circ f \triangleq \theta(\mu_C \circ T\theta^{-1}(g) \circ \theta^{-1}(f))$.

The mono requirement for the monad (T, η, μ) [94] is satisfied because, by Proposition 27 in Section 7.1.2, for each object A the arrow $\eta_A : A \longrightarrow TA$ is an isomorphism $\eta_A = \mathrm{is}_A$ (we denote its inverse by η_A^{-1}), and hence, it is also monic. Consequently, the category **DB** is a *computational model* for the view-mappings (which are the programs) based on observations (i. e., views) with the typed operator T, so that

- TA is a *type of computations* (i. e., observations of the object of values A (of a type A), which are the views of the database A).
- η_A is the *inclusion* of values into computations (i. e., inclusion of elements of the database A into the set of views of the database A). It is an isomorphism $\eta_A = \mathrm{is}_A : A \longrightarrow TA$ in the **DB** category.

– f^* is the *equivalent extension* of a database mapping $f : A \longrightarrow TB$ "from values to computations" (programs correspond to call-by-value parameter passing) to a mapping "from computations to computations" (programs correspond to call-by-name) such that $f^* = Tf = f \circ \eta_A^{-1} = f \circ \mathrm{is}_A^{-1}$ and, consequently, $f^* \approx f$. Thus, in the **DB** category, the call-by-value ($f : A \longrightarrow TB$) and call-by-name ($f^* : TA \longrightarrow TB$) paradigms of programs are represented by *equivalent* morphisms, $f \approx f^*$.
Notice that in skeletal category $\mathbf{DB}_{\mathrm{sk}}$ (which is equivalent to **DB**) all morphisms correspond to the call-by-name paradigm, because each arrow is a mapping from computations into computations (which are closed objects).

However, in our case, the Kleisli category is a perfect model only for a subset of database mappings in **DB**: exactly for every view-mapping (i. e., query) $q_A : A \longrightarrow TA$, which is just an arrow in Kleisli category $\theta(q_A) : A \longrightarrow A$. For a general database mapping $f : A \longrightarrow B$ in **DB**, only its (equivalent to f) "computation extension" $\eta_B \circ f : A \longrightarrow TB$ is an arrow $\theta(\eta_B \circ f) : A \longrightarrow B$ in the Kleisli category.

Consequently, the Kleisli category is a model for the database mappings up to the equivalence "≈."

It means that, generally, the database mappings are not simply programs from values into computations. In fact, the semantics of a database mapping, between any two objects A and B, is equal to telling that for some set of computations (i. e., query-mappings) over A we have the same equivalent (in the sense that these programs produce the same computed value, i. e., a view) set of computations (i. e., query-mappings) over B: it is fundamentally an *equivalence* of computations.

This is a consequence of the fact that each database mapping (which is *a set of functions*) from A into B is naturally bidirectional, i. e., a morphism $f : A \longrightarrow B$ is equivalent to its inverted morphism $f^{\mathrm{OP}} : B \longrightarrow A$ (explained by the duality property $\mathbf{DB} = \mathbf{DB}^{\mathrm{OP}}$ [61]).

Let us define this equivalence formally.

Definition 51. Each simple database mapping $h : A \longrightarrow B$ defines an equivalence of programs (**DB** epimorphisms), $h_A \triangleq \tau(J(h)) : A \twoheadrightarrow TH$ and $h_B \triangleq \tau^{-1}(J(h))^{\mathrm{OP}} : B \twoheadrightarrow TH$, where τ and τ^{-1} are natural transformations of the categorial symmetry in Theorem 14, and H is the kernel (in Definition 67, Section A.3.5) of the information flux $\widetilde{h}$ (i. e., $TH = \widetilde{h}$).

Hence, $h_A \approx h \approx h_B$, such that the computations of these two programs (i. e., arrows of Kleisli category $\mathbf{DB}_T$) are equal, i. e., $\partial_1(h_A) = \partial_1(h_B)$. We can also give an alternative model for the equivalent computational extensions of the database mappings in **DB** category.

Proposition 34. *Denotational semantics of each simple mapping f, between any two database instances A and B, is given by the unique equivalent "computation" arrow $f_1 \triangleq \eta_B \circ f$ in T_{coalg} from the monadic T-coalgebra (A, η_A) into the cofree monadic T-coalgebra (TB, μ_B^C), $f_1 : (A, \eta_A) \longrightarrow (TB, \mu_B^C)$ or, dually, by the unique equivalent ar-*

row $f_1^{\mathrm{OP}} \triangleq (\eta_B \circ f)^{\mathrm{OP}} = f^{\mathrm{OP}} \circ \eta_B^{\mathrm{OP}}$ *from the free monadic T-algebra* (TB, μ_B) *into the monadic T-algebra* $(A, \eta_A^{\mathrm{OP}})$.

Proof. We have to show that for any morphism $f : A \to B$ between an object $A = \biguplus_{1\le j\le m} A_j$, $m \ge 1$ and $B = \biguplus_{1\le i\le k} B_i$, $k \ge 1$, $\mu_B^C \circ \eta_B \circ f = T(\eta_B \circ f) \circ \eta_A$ is valid. In fact, by Proposition 33, η_B is the isomorphism $\mathrm{is}_B : B \to TB$, so that $T\mathrm{is}_B = \mathrm{id}_{TB} : TB \to TB$, due to the fact that

$$\begin{aligned}
\frac{T\mathrm{is}_B}{\biguplus} &= \left\{ T\mathrm{is}_{B_i} : TB_i \to TB_i \mid \mathrm{is}_{B_i} \in \frac{\mathrm{is}_B}{\biguplus} \right\} \quad (\text{from } \widetilde{T\mathrm{is}_{B_i}} = \widetilde{\mathrm{id}_{TB_i}} = TB_i,\ T\mathrm{is}_{B_i} = \mathrm{id}_{TB_i}) \\
&= \left\{ \mathrm{id}_{TB_i} : TB_i \to TB_i \mid \mathrm{is}_{B_i} \in \frac{\mathrm{is}_B}{\biguplus} \right\} \\
&= \{ \mathrm{id}_{TB_i} \mid 1 \le i \le k \} \\
&= \frac{\mathrm{id}_{TB}}{\biguplus},
\end{aligned}$$

and $\mu_B^C = \mathrm{id}_{TB}$. Consequently, $\mu_B^C \circ \eta_B \circ f = \mathrm{id}_{TB} \circ \mathrm{is}_B \circ f = \mathrm{is}_B \circ f$, so that

$$\begin{aligned}
\widetilde{\mu_B^C \circ \eta_B} \circ f \simeq \bigsqcup \widetilde{\frac{\mathrm{is}_B \circ f}{\biguplus}} &= \bigsqcup_{f_{ji} \in \frac{f}{\biguplus}} \widetilde{\mathrm{is}_{B_i} \circ f_{ji}} = \bigsqcup_{f_{ji} \in \frac{f}{\biguplus}} \widetilde{\mathrm{is}_{B_i}} \bigcap \widetilde{f_{ji}} \\
&= \bigsqcup_{f_{ji} \in \frac{f}{\biguplus}} TB_i \bigcap \widetilde{f_{ji}} = \bigsqcup_{f_{ji} \in \frac{f}{\biguplus}} \widetilde{f_{ji}} \simeq \widetilde{f}.
\end{aligned}$$

Analogously, $T(\eta_B \circ f) \circ \eta_A = T\eta_B \circ Tf \circ \mathrm{is}_A = \mathrm{id}_{TB} \circ Tf \circ \mathrm{is}_A = Tf \circ \mathrm{is}_A$. Thus,

$$\begin{aligned}
T(\widetilde{\eta_B \circ f}) \circ \eta_A \simeq \bigsqcup \widetilde{\frac{Tf \circ \mathrm{is}_A}{\biguplus}} &= \bigsqcup_{f_{ji} \in \frac{f}{\biguplus}} \widetilde{Tf_{ji} \circ \mathrm{is}_{A_j}} = \bigsqcup_{f_{ji} \in \frac{f}{\biguplus}} \widetilde{Tf_{ji}} \bigcap \widetilde{\mathrm{is}_{A_j}} \\
&= \bigsqcup_{f_{ji} \in \frac{f}{\biguplus}} \widetilde{Tf_{ji}} \bigcap TA_j = \bigsqcup_{f_{ji} \in \frac{f}{\biguplus}} \widetilde{Tf_{ji}} \simeq \widetilde{Tf} \\
&\simeq \widetilde{f} \quad \text{(by Theorem 11).}
\end{aligned}$$

Thus, $\widetilde{\mu_B^C \circ \eta_B} \circ f \simeq T(\widetilde{\eta_B \circ f}) \circ \eta_A$, so that $\mu_B^C \circ \eta_B \circ f = T(\eta_B \circ f) \circ \eta_A : A \to T^2B$. □

Note that each view-map (i. e., a query) $q_A : A \longrightarrow TA$ is equal to its denotational semantics arrow in T_{coalg}, $q_A : (A, \eta_A) \longrightarrow (TA, \mu_A^C)$.

It is well known that for a Kleisli category there exists an adjunction $(F_T, G_T, \eta_T, \varepsilon_T)$ such that we obtain the same monad (T, η, μ), such that $T = G_T F_T$, $\mu = G_T \varepsilon_T F_T$, $\eta = \eta_T$ (for any arrow $h \in \mathbf{DB}_T(A, B)$, $G_T^0(A) = TA$, $G_T^1(h) = \mu_B \circ T\theta^{-1}(h)$, and for any arrow $f \in \mathbf{DB}(A, B)$, $F_T^0(A) = A$, $F_T^1(f) = \eta_B \circ f$). Let us see now how the Kleisli category $\mathbf{DB}_T$ is "internalized" into the **DB** category.

Theorem 16. *The Kleisli category* $\mathbf{DB}_T$ *of the monad* (T, η, μ) *is isomorphic to* **DB** *category. That is, it may be "internalized" in* **DB** *by the faithful forgetful functor* $K = (K^0, K^1) : \mathbf{DB}_T \longrightarrow \mathbf{DB}$ *such that* K^0 *is an identity function and* $K^1 \triangleq \phi\theta^{-1}$*, where for any two objects A and B, we have the following bijections of the hom-sets:*

- $\theta : \mathbf{DB}(A, TB) \cong \mathbf{DB}_T(A, B)$ *is a Kleisli bijection and*
- $\phi : \mathbf{DB}(A, TB) \cong \mathbf{DB}(A, B)$ *such that* $\phi(_) = \eta^{\mathrm{OP}}_{\mathrm{cod}(_)} \circ (_)$ *is a* **DB** *category bijection, respectively.*

We can generalize a "representation" for the **DB** *category (instead of* **Set** *category):*

A "representation" of functor K is a pair (Υ, φ)*, where* Υ *is the total object and* $\varphi : \mathbf{DB}_T(\Upsilon, _) \simeq K$ *is a natural isomorphism. The functor* $\mathbf{DB}_T(\Upsilon, _) : \mathbf{DB}_T \longrightarrow \mathbf{DB}$ *defines the "internalized" hom-sets in* $\mathbf{DB}_T$*, i. e.,* $\mathbf{DB}^0_T(\Upsilon, B) \triangleq TB^{\Upsilon}$ *and* $\mathbf{DB}^1_T(\Upsilon, f_T) \triangleq \mathrm{id}_\Upsilon \otimes T\theta^{-1}(f_T)$*.*

Proof. Let us show that ϕ is really a bijection in **DB**. For any program morphism $f : A \longrightarrow TB$, we obtain $\phi(f) = \eta^{\mathrm{OP}}_B \circ f : A \longrightarrow B$ and, vice versa, for any $g : A \longrightarrow B$ its inverse $\phi^{-1}(g) \triangleq \eta_B \circ g$; thus, $\phi\phi^{-1}(g) = \phi(\eta_B \circ g) = \eta^{\mathrm{OP}}_B \circ (\eta_B \circ g) = (\eta^{\mathrm{OP}}_B \circ \eta_B) \circ g = \mathrm{id}_B \circ g = g$ (because η_B is an isomorphism), and hence, $\phi\phi^{-1}$ is an identity function. Analogously, $\phi^{-1}\phi(f) = \phi^{-1}(\eta^{\mathrm{OP}}_B \circ f) = \eta_B \circ (\eta^{\mathrm{OP}}_B \circ f) = (\eta_B \circ \eta^{\mathrm{OP}}_B) \circ f = \mathrm{id}_{TB} \circ f = f$, i. e., $\phi^{-1}\phi$ is an identity function, and hence, ϕ is a bijection. Let us demonstrate that K is a functor.

For any identity arrow $\mathrm{id}_T = \theta(\eta_A) : A \longrightarrow A$ in $\mathbf{DB}_T$, we obtain $K^1(\mathrm{id}_T) = \phi\theta^{-1}(\theta(\eta_A)) = \phi(\eta_A) = \eta^{\mathrm{OP}}_A \circ \eta_A = \mathrm{id}_A$ (because η_A is an isomorphism). For any two arrows $g_T : B \longrightarrow C$ and $f_T : A \longrightarrow B$ in Kleisli category, we obtain

$$K^1(g_T \circ f_T) = K^1\big(\theta(\mu_C \circ T\theta^{-1}(g_T) \circ \theta^{-1}(f_T))\big)$$
$$\text{(from Definition 50 of the Kleisli category);}$$

we introduce

$$\begin{aligned}
&g \triangleq \theta^{-1}(g_T) : B \longrightarrow TC \text{ and } f \triangleq \theta^{-1}(f_T) : A \longrightarrow TB\\
&= \phi\theta^{-1}(\theta(\mu_C \circ Tg \circ f)) = \phi\theta^{-1}(\theta(\mathrm{id}_{TC} \circ Tg \circ f)) = \phi\theta^{-1}(\theta(Tg \circ f))\\
&\quad\left(\text{it is easy to verify in } \mathbf{DB} \text{ that } Tg = g \circ \eta^{\mathrm{OP}}_B, \text{ from } \widetilde{Tg} \simeq \widetilde{g} \simeq \left\lfloor + \right\rfloor \widetilde{\frac{g \circ \eta^{\mathrm{OP}}_B}{\left\lfloor + \right\rfloor}}\right)\\
&= \phi(g \circ \eta^{\mathrm{OP}}_B \circ f) = \eta^{\mathrm{OP}}_C \circ g \circ \eta^{\mathrm{OP}}_B \circ f = \phi(g) \circ \phi(f) = \phi\theta^{-1}(\theta(g)) \circ \phi\theta^{-1}(\theta(f))\\
&= K^1(\theta\theta^{-1}(g_T)) \circ K^1(\theta\theta^{-1}(f_T)) = K^1(g_T) \circ K^1(f_T).
\end{aligned}$$

Thus, each arrow $f_T : A \longrightarrow B$ in $\mathbf{DB}_T$ is "internalized" in **DB** by its representation $f \triangleq K^1(f_T) = \phi\theta^{-1}(f_T) = \eta^{\mathrm{OP}}_B \circ \theta^{-1}(f_T) : A \longrightarrow B$, where $\theta^{-1}(f_T) : A \longrightarrow TB$ is a program equivalent to the database mapping $f : A \longrightarrow B$, i. e., $\theta^{-1}(f_T) \approx f$.

The functor K is faithful. In fact, for any two arrows $f_T, h_T : A \longrightarrow B$ in $\mathbf{DB}_T$, $K^1(f_T) = K^1(h_T)$ implies $f_T = h_T$. Indeed, from $K^1(f_T) = K^1(h_T)$ we obtain $\phi\theta^{-1}(f_T) = \phi\theta^{-1}(h_T)$, and if we apply a bijection $\phi\theta^{-1}$, we obtain $\phi\theta^{-1}\phi\theta^{-1}(f_T) = \phi\theta^{-1}\phi\theta^{-1}(h_T)$, i. e., $\theta\theta^{-1}(f_T) = \theta\theta^{-1}(h_T)$, i. e., $f_T = h_T$ (the compositions $\theta\theta^{-1}$ and $\phi\phi^{-1}$ are the identity functions).

Let us show that K is an isomorphism: from the adjunction $(F_T, G_T, \eta_T, \varepsilon_T) : \mathbf{DB} \longrightarrow \mathbf{DB}_T$, where F_T^0 is identity, $F_T^1 \triangleq \theta\phi^{-1}$, we obtain that $F_T \circ K = I_{DB_\top}$ and $K \circ F_T = I_{\mathrm{DB}}$. Thus, the functor K is an isomorphism of **DB** and Kleisli category $\mathbf{DB}_T$.

Notice that from Theorem 18 in Section 7.4.1, $\mathbf{DB}_T^0(\mathrm{Y}, A) \triangleq TA^{\mathrm{Y}} = A^{\mathrm{Y}} = TA$ and $\mathbf{DB}_T^1(\mathrm{Y}, f_T) \triangleq \mathrm{id}_{\mathrm{Y}} \otimes T\theta^{-1}(f_T) = T\theta^{-1}(f_T)$.

Thus, $K^0(A) = A \simeq TA = \mathbf{DB}_T^0(\mathrm{Y}, A)$, and for $f_T : A \to B$, $K^1(f_T) = \phi\theta^{-1}(f_T) = \eta^{\mathrm{OP}} \circ \theta^{-1}(f_T) : A \to B$, and from the fact that $Tg \approx \mathrm{is}^{-1}_{\mathrm{cod}(g)} \circ g = \eta^{\mathrm{OP}}_{\mathrm{cod}(g)} \circ g$ for any arrow g in **DB**, we obtain $K^1(f_T) \approx \mathbf{DB}_T^1(\mathrm{Y}, f_T)$. Consequently, we obtain the natural isomorphism $\varphi : \mathbf{DB}_T(\mathrm{Y}, _) \simeq K$. □

Remark. It is easy to verify that a natural isomorphism $\eta : I_{\mathrm{DB}} \stackrel{\cdot}{\longrightarrow} T$ of the monad (T, η, μ) is equal to the natural transformation $\eta : K \stackrel{\cdot}{\longrightarrow} G_T$ (by considering that $G_T : \mathbf{DB}_T \longrightarrow \mathbf{DB}$ is defined by $G_T^0 = T^0$ and for any $f_T : A \longrightarrow B$ in $\mathbf{DB}_T$, $G_T^1(f_T) \triangleq \mu_B \circ T\theta^{-1}(f_T) : TA \longrightarrow TB$).

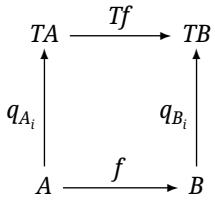
Hence, the functor F_T has two different adjunctions: the universal adjunction $(F_T, G_T, \eta_T, \mu_T)$, which defines the same monad (T, η, μ), and this particular (for the **DB** category only) isomorphism's adjunction (F_T, K, η_I, μ_I), which defines the trivial identity monad. □

We are now ready to define the semantics of queries in **DB** category and the categorial definition of *query equivalence*. This is important in the context of the database integration/exchange and for the theory of *query-rewriting* [100].

When we define a mapping (i. e., an arrow or morphism) $f : A \longrightarrow B$ between two simple databases A and B, we define implicitly the information flux $\widetilde{f}$, i. e., the set of views of A transferred by this mapping into B. Thus, in the context of the query rewriting, we consider only the queries (i. e., view maps) whose resulting view (i. e., an observation) belongs to the information flux of this mapping. Consequently, given any two queries, $q_{A_i} : A \longrightarrow TA$ and $q_{B_j} : B \longrightarrow TB$, they have to satisfy (w. r. t. query rewriting constraints) the condition $\partial_1(q_{A_i}) \in \widetilde{f}$ (the $\partial_1(q_{A_i})$ is a singleton containing the resulting view of this query) and $\partial_1(q_{B_j}) \in \widetilde{f}$. So, a well-rewritten query over B, $q_{B_j} : B \longrightarrow TB$, such that it is equivalent to the original query, i. e., $q_{B_j} \approx q_{A_i}$, has to satisfy the condition $\partial_1(q_{B_j}) = \partial_1(q_{A_i}) \in \widetilde{f}$. Now we can provide the denotational semantics for a query-rewriting in a data integration/exchange environment.

Proposition 35. *Each database query is a (nonmonadic) T-coalgebra. Any homomorphism between two T-coalgebras $f : (A, q_{A_i}) \longrightarrow (B, q_{B_j})$, where A and B are two simple objects (databases) in* **DB**, *defines the semantics for the relevant query rewriting, when $\partial_1(q_{A_i}) \in \widetilde{f}$.*

Proof. Let us consider the following commutative diagram, where vertical arrows are the T-coalgebras:



The homomorphism between two (nonmonadic) T-coalgebras $f : (A, q_{A_i}) \longrightarrow (B, q_{B_j})$ means that the commutativity $q_{B_j} \circ f = Tf \circ q_{A_i} : A \longrightarrow TB$ is valid and, from the duality property, we obtain $q_{B_i} = Tf \circ q_{A_i} \circ f^{\mathrm{OP}}$. Consequently, for a given mapping $f : A \to B$ between databases A and B, every query q_{A_i} such that $\partial_1(q_{A_i}) \in \widetilde{f}$ (i. e., $\widetilde{q_{A_i}} \subseteq \widetilde{f}$), we have an equivalent rewritten query q_{B_i} over a database B. In fact, $\widetilde{q_{B_i}} = \widetilde{Tf} \bigcap \widetilde{q_{A_i}} \bigcap \widetilde{f^{\mathrm{OP}}} = \widetilde{q_{A_i}}$ because of the fact that $\widetilde{q_{A_i}} \subseteq \widetilde{f}$ and $\widetilde{f^{\mathrm{OP}}} = \widetilde{Tf} = \widetilde{f}$. Consequently, we obtain the equivalence of these two queries $q_{B_j} \approx q_{A_i}$. □

7.4 Partial ordering for databases

Let us consider the "observational" Partial Order (PO) relation introduced in point 3 of Definition 40 in Section 7.1, its extension to morphisms and a 2-category property of **DB**.

Theorem 17. *PO subcategory* $\mathbf{DB}_I \subseteq \mathbf{DB}$ *is defined by* $\mathrm{Ob}_{DB_I} = \mathrm{Ob}_{\mathrm{DB}}$ *and* Mor_{DB_I} *is a union of all isomorphisms and initial arrows* $\langle \overbrace{\perp^1, \ldots, \perp^1}^{m} \rangle : \perp^0 \hookrightarrow \biguplus_{1 \le j \le m} A_j$ *and* $S_I = \{\mathrm{in} : A \hookrightarrow B \mid$ *for two strictly-complex objects* $A = \biguplus_{1 \le j \le m} A_j \preceq B = \biguplus_{1 \le i \le k} B_i$ *with a mapping (from Definition* 40*)* $\sigma : \{1, \ldots, m\} \to \{1, \ldots, k\}$ *and* $\overset{in}{\biguplus} = \{\mathrm{in}_{j\sigma(j)} : A_j \hookrightarrow B_{\sigma(j)} \mid j = 1, \ldots, m\}\}$.

This ordering can be extended by categorial symmetry property to morphisms as well:

$$f \preceq g \quad \textit{iff} \quad \widetilde{f} \preceq \widetilde{g}$$

The power-view endofunctor $T : \mathbf{DB} \longrightarrow \mathbf{DB}$ *is a 2-endofunctor and a closure operator.*

DB *is a 2-category and 1-cells are its ordinary morphisms, while 2-cells (denoted by* $\sqrt{_}$*) are the arrows between ordinary morphisms: for any two morphisms* $f, g : A \longrightarrow B$ *such that* $f \preceq g$*, a 2-cell arrow is the 'inclusion"* $\sqrt{\beta} : f \xrightarrow{\preceq} g$.

Such a 2-cell arrow is represented by an ordinary monomorphism in **DB**, $\beta : \widetilde{f} \hookrightarrow \widetilde{g}$, *with* $\beta = T_e^1(f_{\mathrm{ep}}^{\mathrm{OP}} \circ f_{\mathrm{ep}}; f_{in} \circ f_{in}^{\mathrm{OP}})$, *where* $f_{\mathrm{ep}} = \tau(J(f)) : A \twoheadrightarrow \widetilde{f}$ *is an epimorphism, and the arrow* $f_{in} = \tau^{-1}(J(f)) : \widetilde{f} \hookrightarrow B$ *is a monomorphism.*

Proof. $\mathbf{DB}_I$ is well-defined: for any object A we have that its identity arrow $\mathrm{id}_A : A \to A$ is a monomorphism (we have $A \preceq A$ with σ identity function, and $in = \mathrm{id}_A$), thus an arrow in $\mathbf{DB}_I$. Each isomorphism is a monomorphism as well, and any initial arrow is monic as well, so that all arrows and their compositions in $\mathbf{DB}_I$ are monic arrows. For each isomorphism $A \simeq B$ represented by arrow $\mathrm{is} : A \to B$ in $\mathbf{DB}_I$, we define $\sigma : \{1,\ldots,m\} \to \{1,\ldots,k\}$ such that $\sigma(j) = i$ if $(\mathrm{is}_{ji} : A_j \to B_i) \in \frac{\mathrm{is}}{\biguplus} = \perp^0$; 1 otherwise (when $A_j = \perp^0$). Thus for each $j = 1,\ldots,m$, $TA_j \subseteq TB_{\sigma(j)}$, hence $A \preceq B$.

For each initial arrow $\langle \overbrace{\perp^1,\ldots,\perp^1}^{m} \rangle : \perp^0 \hookrightarrow \biguplus_{1\le j\le m} A_j$ and any $\sigma : \{1\} \to \{1,\ldots,m\}$, $T\ \perp^0 = \perp^0 \subseteq TA_{\sigma(1)}$; thus; it represents the PO relation $\perp^0 \preceq A$. Any monomorphism in S_I represents a PO relation $A \preceq B$ between two strictly-complex objects. Each composed arrow by the monic arrows above is composition of PO relations of each of its components, and hence, a PO relation as well. Consequently, each arrow f in $\mathbf{DB}_I$ represents the PO relation $\mathrm{dom}(f) \preceq \mathrm{cod}(f)$.

Let us show that each PO relation $A \preceq B$ is represented by a monomorphism in $\mathbf{DB}_I$ as well: if $A \simeq \perp^0$, then it is represented by a composition of the isomorphism (monic as well) is $: A \to \perp^0$ and initial arrow from $\perp^0$ into B, which is monic as well. Otherwise $A \preceq B$ is generally representable as $A \simeq A_S \preceq B_S \simeq B$ where A_S is a strictly-complex object obtained by eliminating all objects $\perp^0$ from A (analogously for B_S). Thus, $A \preceq B$ is represented in $\mathbf{DB}_I$ as composition of these two isomorphisms (that are also monic) and a monomorphism in S_I between two strictly-complex objects (the composition of monic arrows is monic as well).

Consequently, between any two given objects in $\mathbf{DB}_I$ there can exist at most one arrow, so this is a PO category. Power-view operator T satisfies the following:

(1) $TA = T(TA)$, as explained in the introduction, Section A.3.2.
(2) $A \subseteq B$ implies $TA \subseteq TB$: T is a monotonic operator w. r. t. $\subseteq$ (the free SPRJU algebra of terms $\mathcal{L}_A$ is monotonic with respect to its carrier set A; see Section A.3.2).
(3) $A \subseteq TA$, each element of A is also a view of A.

Thus, T is a closure operator and an object A such that $A = TA$ is a closed object.

For any two simple morphisms $f, g : A \to B$ such that $\widetilde{f} \subseteq \widetilde{g}$, it is easy to show that there is the monomorphism $\beta = T^1_e(f^{\mathrm{OP}}_{\mathrm{ep}} \circ f_{\mathrm{ep}};\ f_{in} \circ f^{\mathrm{OP}}_{in}) : \widetilde{f} \hookrightarrow \widetilde{g}$. In fact, for $h_1 = f^{\mathrm{OP}}_{\mathrm{ep}} \circ f_{\mathrm{ep}} : A \to A$ with $\widetilde{h_1} = \widetilde{f^{\mathrm{OP}}_{\mathrm{ep}}} \bigcap \widetilde{f_{\mathrm{ep}}} = \widetilde{f} \bigcap \widetilde{f} = \widetilde{f}$ and $h_2 = f_{in} \circ f^{\mathrm{OP}}_{in} : B \to B$ with $\widetilde{h_2} = \widetilde{f_{in}} \bigcap \widetilde{f^{\mathrm{OP}}_{in}} \widetilde{f} \bigcap \widetilde{f} = \widetilde{f}$, we obtain that $\widetilde{h_2 \circ f} = \widetilde{f} \bigcap \widetilde{f} = \widetilde{f}$ and $\widetilde{g \circ h_1} = \widetilde{g} \bigcap \widetilde{f} = \widetilde{f}$ (for $\widetilde{f} \subseteq \widetilde{g}$), so that $h_2 \circ f = g \circ h_1$ and, consequently, the morphism $T^1_e(h_1; h_2) = T^1_e(f^{\mathrm{OP}}_{\mathrm{ep}} \circ f_{\mathrm{ep}};\ f_{in} \circ f^{\mathrm{OP}}_{in}) : \widetilde{f} \to \widetilde{g}$ is well-defined. Thus, $\beta = T^1_e(f^{\mathrm{OP}}_{\mathrm{ep}} \circ f_{\mathrm{ep}};\ f_{in} \circ f^{\mathrm{OP}}_{in}) = \mathrm{in}^{\mathrm{OP}}_{\widetilde{g}} \circ T(f_{in} \circ f^{\mathrm{OP}}_{in} \circ f) \circ \mathrm{in}_{\widetilde{f}}$, so that (from $\widetilde{Tf^{\mathrm{OP}}_{in}} = \widetilde{Tf_{in}} = \widetilde{Tf} = \widetilde{f}$), $\widetilde{\beta} = \widetilde{\mathrm{in}^{\mathrm{OP}}_{\widetilde{g}}} \bigcap \widetilde{Tf_{in}} \bigcap \widetilde{Tf^{\mathrm{OP}}_{in}} \bigcap \widetilde{Tf} \bigcap \widetilde{\mathrm{in}_{\widetilde{f}}} = \widetilde{g} \bigcap \widetilde{f} = \widetilde{f}$, so that $\beta = T^1_e(f^{\mathrm{OP}}_{\mathrm{ep}} \circ f_{\mathrm{ep}};\ f_{in} \circ f^{\mathrm{OP}}_{in})$ is a monomorphism (from Corollary 17).

It is easy to verify that **DB** is a 2-category with 0-cells (its objects), 1-cells (its ordinary morphisms) and 2-cells ("inclusions" arrows between mappings). The horizontal and vertical composition of 2-cells is just the composition of PO relations. Given $f, g, h : A \longrightarrow B$ with 2-cells $\sqrt{\beta} : f \xrightarrow{\preceq} g$, $\sqrt{\delta} : g \xrightarrow{\preceq} h$, their vertical composition is $\sqrt{\gamma} = \sqrt{\delta} \circ \sqrt{\beta} : f \xrightarrow{\preceq} h$. Given $f, g : A \longrightarrow B$ and $h, l : B \longrightarrow C$ with 2-cells $\sqrt{\beta} : f \xrightarrow{\preceq} g$, $\sqrt{\delta} : h \xrightarrow{\preceq} l$, for a given composition functor $\bullet : \mathbf{DB}(A,B) \times \mathbf{DB}(B,C) \longrightarrow \mathbf{DB}(A,C)$, their horizontal composition is $\sqrt{\gamma} = \sqrt{\delta} \bullet \sqrt{\beta} : h \circ f \xrightarrow{\preceq} l \circ g$. □

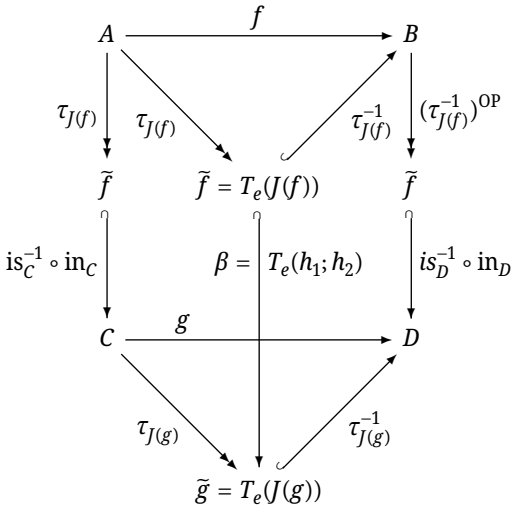
For example, the equivalence $q_{A_i} \approx q_{B_j}$ (i. e., $q_{A_i} \preceq q_{B_j}$ and $q_{B_j} \preceq q_{A_i}$) of two view mappings $q_{A_i} : A \longrightarrow TA$ and $q_{B_j} : B \longrightarrow TB$, for the simple databases A and B, is obtained when they produce the same view.

Remark. Note the analogy of the 2-category **DB** and the much more simple 2-category **Z** of finite labeled trees, specified by Definition 31 and Proposition 15 in Section 6.3. A morphism in **DB** is generally a set of trees (see Figure 7.2 in Section 7.1) while in **Z** a morphism is a single tree: so, from this point of view, the **DB** category generalizes (the arcs of trees of a morphism in **DB** are *functions* and not simple labels as in **Z**) the more simple category **Z**. □

The categorial symmetry operator $T_e^0 J : \mathrm{Mor}_{\mathrm{DB}} \longrightarrow \mathrm{Ob}_{\mathrm{DB}}$ for any morphism f in **DB** produces its information flux $\widetilde{f}$ (i. e., the "conceptualized" database of this mapping). Consequently, we can define a "mapping between mappings" (which are 2-cells "inclusions") and also all higher n-cells [101], with their direct representation by a 1-cell morphisms.

Example 28. Let us consider the two ordinary simple morphisms (1-cells) in **DB**, $f : A \longrightarrow B$ and $g : C \longrightarrow D$ such that $\widetilde{f} \subseteq \widetilde{g}$ (thus, $f \preceq g$) and $\widetilde{g} \subseteq TC \bigcap TD$. We want to show that the 1-cells corresponding monomorphism $\beta : \widetilde{f} \hookrightarrow \widetilde{g}$ is a result of the symmetric closure functor T_e. We define two arrows $h_1 = \mathrm{is}_C^{-1} \circ \mathrm{in}_C \circ \tau(J(f))$ and $h_2 = \mathrm{is}_D^{-1} \circ \mathrm{in}_D \circ (\tau_{J(f)}^{-1})^{\mathrm{OP}}$ (where $\mathrm{in}_C : \widetilde{f} \hookrightarrow TC$ is a monomorphism, well-defined in Proposition 24 because $\widetilde{f} \subseteq \widetilde{g} \subseteq TC$, $\mathrm{is}_C^{-1} : TC \longrightarrow C$ is an isomorphism and $\mathrm{in}_D : \widetilde{f} \hookrightarrow TD$ is a monomorphism from Proposition 24 and $\widetilde{f} \subseteq \widetilde{g} \subseteq TD$, and $\mathrm{is}_D^{-1} : TD \longrightarrow D$ is an isomorphism). Hence, $\widetilde{h_1} = \widetilde{\mathrm{is}_C^{-1}} \bigcap \widetilde{\mathrm{in}_C} \bigcap \widetilde{\tau(J(f))} = TC \bigcap T\widetilde{f} \bigcap \widetilde{f} = \widetilde{f}$ (because $TC \supseteq \widetilde{g} \supseteq \widetilde{f}$ and $T\widetilde{f} = \widetilde{f}$) and, analogously, $\widetilde{h_2} = \widetilde{\mathrm{is}_D^{-1}} \bigcap \widetilde{\mathrm{in}_D} \bigcap \widetilde{(\tau_{J(f)}^{-1})^{\mathrm{OP}}} = TD \bigcap T\widetilde{f} \bigcap \widetilde{f} = \widetilde{f}$ (because $TD \supseteq \widetilde{g} \supseteq \widetilde{f}$ and $T\widetilde{f} = \widetilde{f}$). Thus, $\widetilde{g \circ h_1} = \widetilde{h_2 \circ f} = \widetilde{f}$, and hence, $g \circ h_1 = h_2 \circ f$.

Thus, there exists the arrow $\beta = T_e(h_1; h_2) : J(f) \longrightarrow J(g)$ in $\mathbf{DB} \downarrow \mathbf{DB}$, and the following commutative diagram in **DB**:



Let us show that also $T_e(h_1; h_2)$ is a monomorphism with $\beta = T_e(h_1; h_2) : \widetilde{f} \hookrightarrow \widetilde{g}$. In fact, $\widetilde{h_2} = \widetilde{f}$ and, by definition in Theorem 14, $T_e(h_1; h_2) \triangleq \mathrm{in}^{\mathrm{OP}}_{\widetilde{g}} \circ T(h_2 \circ f) \circ \mathrm{in}_{\widetilde{f}}$. Thus, $\widetilde{T_e(h_1; h_2)} = \widetilde{\mathrm{in}^{\mathrm{OP}}_{\widetilde{g}}} \bigcap \widetilde{Th_2} \bigcap \widetilde{Tf} \bigcap \mathrm{in}_{\widetilde{f}} = \widetilde{g} \bigcap \widetilde{f} = \widetilde{f}$ and, consequently, $T_e(h_1; h_2)$ is a monomorphism (from Corollary 17). In the particular case of Theorem 17 when $A = C$ and $B = D$ and $\mathrm{is}^{-1}_C \circ \mathrm{in}_C = (\tau_{J(f)})^{\mathrm{OP}} = f^{\mathrm{OP}}_{\mathrm{ep}}$ and $\mathrm{is}^{-1}_D \circ \mathrm{in}_D = \tau^{-1}_{J(f)} = f_{in}$, we obtain that the 2-cell's arrow $\sqrt{\beta} : f \xrightarrow{\preceq} g$ is represented by the 1-cell monomorphism $\beta = T^1_e(f^{\mathrm{OP}}_{\mathrm{ep}} \circ f_{\mathrm{ep}}; f_{in} \circ f^{\mathrm{OP}}_{in}) : \widetilde{f} \hookrightarrow \widetilde{g}$.

Let us define the full subcategory of **DB** composed of only simple objects.

Definition 52. The full subcategory of **DB** composed of only simple objects will be denoted as $\underline{\mathbf{DB}}$, with the power-view endofunctor equal to the restriction of T to simple objects and arrows only.

The bottom object in $\underline{\mathbf{DB}}$ is that of **DB**, equal to $\perp^0$ (zero object), while the top object is equal to $\underline{\Upsilon} = \bigcup_{A \in \mathrm{Ob}_{\underline{\mathbf{DB}}}} A \subset \Upsilon = \bigcup_{A \in \mathrm{Ob}_{\mathbf{DB}}} A$.

The exact relationship between $\underline{\Upsilon}$ and the top complex object Υ is given by

$$\Upsilon = \underline{\Upsilon} \bigcup \left(\biguplus_{\omega} \underline{\Upsilon} \right) = \underline{\Upsilon} \bigcup \overbrace{\left(\underline{\Upsilon} \uplus \underline{\Upsilon} \uplus \cdots \right)}^{\omega}.$$

Let us show that the objects Υ and $\perp^0$ are the top and bottom objects in **DB** w. r. t. the PO relation $\preceq$.

Proposition 36. *For any object $A \in \mathrm{Ob}_{\mathrm{DB}}$ (for $A = \Upsilon$ as well), the following is valid:*

1. $A \otimes \Upsilon \preceq \Upsilon$ *and* $A \uplus \Upsilon \preceq \Upsilon$.
2. Υ *and* $\perp^0$ *are the top and bottom objects, respectively, (up to isomorphism) in* **DB**.

Proof. 1. From Definition 40, we have

$$A \otimes \Upsilon = \biguplus_{1\le j\le m \,\&\, i=1,2,\ldots} TA_j \bigcap \Upsilon_i = \biguplus_{1\le j\le m \,\&\, i=1,2,\ldots} TA_j \bigcap \underline{\Upsilon}$$
$$= \biguplus_{1\le j\le m \,\&\, i=1,2,\ldots} TA_j = \left(TA_1 \biguplus \cdots \biguplus TA_m\right) \biguplus \left(TA_1 \biguplus \cdots \biguplus TA_m\right) \biguplus \cdots$$

Thus, we have an identity mapping $\sigma : \{1,2,\ldots\} \to \{1,2,\ldots\}$ such that $T(A \otimes \Upsilon)_j \in \{A_1,\ldots,A_m\}$ and $T\Upsilon_{\sigma(j)} = T\underline{\Upsilon}$ so that $T(A \otimes \Upsilon)_j \subseteq T\Upsilon_{\sigma(j)}$ for all $j \ge 1$, and hence, $A \otimes \Upsilon \preceq \Upsilon$.

Hence, $A \biguplus \Upsilon = A_1 \biguplus \cdots \biguplus A_m \biguplus \Upsilon$ and for the identity mapping $\sigma : \{1,2,\ldots\} \to \{1,2,\ldots\}$, for $1 \le j \le m$, $T(A \biguplus \Upsilon)_j = TA_j \subseteq T\Upsilon\sigma(j) = T\underline{\Upsilon}$, and for $j > m$, $T(A \biguplus \Upsilon)_j = T\underline{\Upsilon} \subseteq T\Upsilon\sigma(j) = T\underline{\Upsilon}$. Thus, $A \biguplus \Upsilon \preceq \Upsilon$.

2. For any object A in **DB**, $TA \subseteq \bigcup_{B\in \mathrm{Ob}_{\mathrm{DB}}} B = \Upsilon = T\Upsilon$. Thus, $A \preceq \Upsilon$ and $TA \preceq \Upsilon$, i. e., the closed object Υ is a top object. Notice that from point 2 of Proposition 24 there exist a monomorphism $\mathrm{in}_{TA} : TA \hookrightarrow \Upsilon$, and $\mathrm{in}_{TA} \circ \mathrm{is}_A : A \hookrightarrow \Upsilon$ is a composition of two monomorphisms ($\mathrm{is}_A : A \to TA$ is an isomorphism, thus the monic arrow as well), and hence, a monomorphism as well.

From the fact that $\perp \in A$ for any object A in **DB** and the fact that $\perp^0 = \{\perp\}$, $\perp^0 \subseteq A$, so that $\perp^0 \preceq A$, we obtain that $\perp^0$ is a bottom object. □

7.4.1 Matching tensor product

Since the data residing in different databases may have interdependencies (they are based on the partial overlapping between databases, which is information about a common part of the world), we can define such an (partial) overlapping by morphisms of the category **DB**: information flux of each mapping between two objects A and B in **DB** is just a subset of this overlapping between these two databases, denoted by $A \otimes B$. It is "bidirectional," i. e., (by duality) for any mapping f from A into B there exists an *equivalent* mapping f^{OP} from B into A.

This overlapping represents the common matching between these two databases, and is equal to the maximal information flux, which can be defined between these two databases. Consequently, we can formally introduce a denotational semantics for the database matching operation $\otimes$ as follows.

Theorem 18. **DB** *is a strictly symmetric idempotent monoidal category* $(\mathbf{DB}, \otimes, \Upsilon, \alpha, \beta, \gamma)$, *where* Υ *is the total object for a given universe of databases, with the "matching" tensor product* $\otimes : \mathbf{DB} \times \mathbf{DB} \longrightarrow \mathbf{DB}$ *defined as follows:*

1. $\Upsilon \otimes A = A \otimes \Upsilon \triangleq TA$;
   Otherwise, for any two database instances (objects) $A = \biguplus_{1\le j\le m} A_j$ *and* $B = \biguplus_{1\le i\le k} B_i$, *where all* A_j *and* B_i *are simple objects, with* $k, m \ge 1$, *the object* $A \otimes B \equiv \otimes(A, B)$ *is the overlapping (matching) between A and B, introduced by Definition* 40*:*

$$A \otimes B \triangleq \begin{cases} TA \bigcap TB, & \text{if } m = k = 1, A = A_1, B = B_1; \\ \biguplus_{1 \le i \le k \,\&\, 1 \le j \le m} (A_j \otimes B_i), & \text{otherwise.} \end{cases}$$

2. $\mathrm{id}_Y \otimes f = f \otimes \mathrm{id}_Y \triangleq Tf$;
Otherwise, for any two arrows $f : A \to C$ and $g : B \to D$, the $f \otimes g \equiv \otimes(f, g) : A \otimes B \to C \otimes D$ is defined by canonical representation (Lemma 2) of the set of ptp arrows

$$\frac{f \otimes g}{\biguplus} \triangleq \begin{cases} \{\{\mathrm{id}_R : R \to R \mid R \in (\widetilde{f} \bigcap \widetilde{g}) \neq \perp^0\}\}, & \text{if } \frac{f}{\biguplus} = \{f\}, \frac{g}{\biguplus} = \{g\}; \\ \{h_{ij} = \{\mathrm{id}_R \mid R \in \widetilde{f_i} \bigcap \widetilde{g_j}\} \neq \perp^1 \mid f_i \in \frac{f}{\biguplus}, g_j \in \frac{g}{\biguplus}\}, & \text{otherwise,} \end{cases}$$

such that for each point-to-point arrow $h_{ij} : T(\mathrm{dom}(f_i)) \bigcap T(\mathrm{dom}(g_j)) \to T(\mathrm{cod}(f_i)) \bigcap T(\mathrm{cod}(g_j))$ in $\frac{f \otimes g}{\biguplus}$, one has $\widetilde{h_{ij}} = \widetilde{f_i} \bigcap \widetilde{g_j} \neq \perp^0$.

Proof. Let us show that for any two simple arrows $f : A \to C$ and $g : B \to D$ in **DB** category, also the arrow $f \otimes g$ is a **DB** category arrow (i. e., it satisfies Theorem 10).

From the fact that f and g are the arrows in **DB**, it holds from Theorem 10 that there exist the schema mappings $\mathcal{M}_{AC} : \mathcal{A} \to \mathcal{C}$, $\mathcal{M}_{BD} : \mathcal{B} \to \mathcal{D}$, and an R-algebra α, such that $f = \alpha^*(\mathrm{MakeOperads}(\mathcal{M}_{AC}))$ and $g = \alpha^*(\mathrm{MakeOperads}(\mathcal{M}_{BD}))$ with $A = \alpha^*(\mathcal{A})$, $B = \alpha^*(\mathcal{B})$, $C = \alpha^*(\mathcal{C})$ and $D = \alpha^*(\mathcal{D})$.

Hence, we can specify the α-intersection schema mapping (from Definition 70),

$$\mathcal{M}_{AC} \underset{\alpha}{\bigcap} \mathcal{M}_{BD} = \{\Phi\} : \mathcal{A} \underset{\alpha}{\bigcap} \mathcal{B} \to \mathcal{C} \underset{\alpha}{\bigcap} \mathcal{D}, \quad \text{where } \Phi \text{ is a tgd}$$
$$\bigwedge \Big\{ \forall \mathbf{x}(r(\mathbf{x}) \Rightarrow r(\mathbf{x})) \mid r \in \mathcal{A} \underset{\alpha}{\bigcap} \mathcal{B}, \alpha(r) \in \mathrm{Flux}(\alpha, \mathrm{MakeOperads}(\mathcal{M}_{AC})) \bigcap \mathrm{Flux}(\alpha, \mathrm{MakeOperads}(\mathcal{M}_{BD})) \Big\}.$$

Consequently, $\mathbf{M} = \mathrm{MakeOperads}(\mathcal{M}_{AC} \underset{\alpha}{\bigcap} \mathcal{M}_{BD}) = \{1_r \mid r \in \mathcal{A} \underset{\alpha}{\bigcap} \mathcal{B}, \alpha(r) \in \mathrm{Flux}(\alpha, \mathrm{MakeOperads}(\mathcal{M}_{AC})) \bigcap \mathrm{Flux}(\alpha, \mathrm{MakeOperads}(\mathcal{M}_{BD}))\} \bigcup \{1_{r_\emptyset}\} = \{1_r \mid r \in \mathcal{A} \underset{\alpha}{\bigcap} \mathcal{B}, \alpha(r) \in \widetilde{f} \bigcap \widetilde{g}\} \bigcup \{1_{r_\emptyset}\}$ is a sketch category arrow from the schema $\mathcal{A} \underset{\alpha}{\bigcap} \mathcal{B}$ into the schema $\mathcal{B} \underset{\alpha}{\bigcap} \mathcal{D}$, with $\alpha^*(\mathcal{A} \underset{\alpha}{\bigcap} \mathcal{B}) = TA \bigcap TB = A \otimes B$, $\alpha^*(\mathcal{C} \underset{\alpha}{\bigcap} \mathcal{D}) = TC \bigcap TD = C \otimes D$, so that

$$\alpha^*(\mathbf{M}) = \Big\{ \alpha(1_r) = \mathrm{id}_R : R \to R \mid r \in \mathcal{A} \underset{\alpha}{\bigcap} \mathcal{B}, R = \alpha(r) \in \widetilde{f} \bigcap \widetilde{g} \Big\} \bigcup \{q_\perp\} = f \otimes g : A \otimes B \to C \otimes D.$$

Let us show that for any four simple arrows $f : A \to C, f' : C \to E, g : B \to D$ and $g' : B \to F$ (from Definition 43 in Section 7.1, $\frac{f}{\biguplus} = \{f\}$, and analogously for all other arrows), where all A, B, C, D, E and F are simple objects, $(f' \circ f) \otimes (g' \circ g) = (f' \otimes g') \circ (f \otimes g) : A \otimes B \to E \otimes F$. Thus, $\widetilde{(f' \circ f) \otimes (g' \circ g)} = T\widetilde{(f' \circ f)} \bigcap T\widetilde{(g' \circ g)} =$

$T(\widetilde{f'} \bigcap \widetilde{f}) \bigcap T(\widetilde{g'} \bigcap \widetilde{g}) = (\widetilde{\widetilde{f'} \bigcap \widetilde{f}}) \bigcap (\widetilde{g'} \bigcap \widetilde{g}) = (\widetilde{f'} \bigcap \widetilde{g'}) \bigcap (\widetilde{f} \bigcap \widetilde{g}) = T(\widetilde{f'} \bigcap \widetilde{g'}) \bigcap T(\widetilde{f} \bigcap \widetilde{g}) = \widetilde{f' \otimes g'} \bigcap \widetilde{f \otimes g} = (f' \otimes g') \circ (f \otimes g)$, so that from Definition 44,
(1) $(f' \circ f) \otimes (g' \circ g) = (f' \otimes g') \circ (f \otimes g)$.

Let us show that this property of composition is valid for complex morphisms $f : A \to C$, $f' : C \to E$, $g : B \to D$ and $g' : B \to F$ as well. Then

$$\begin{aligned}
\frac{(f' \otimes g') \circ (f \otimes g)}{\biguplus} &= \left\{ (f'_i \otimes g'_j) \circ (f_i \otimes g_j) \mid f'_i \otimes g'_j \in \frac{f' \otimes g'}{\biguplus}, f_i \otimes g_j \in \frac{f \otimes g}{\biguplus}, \right. \\
&\qquad \left. \operatorname{cod}(f_i \otimes g_j) = \operatorname{dom}(f'_i \otimes g'_j) \right\} \\
&= \left\{ (f'_i \otimes g'_j) \circ (f_i \otimes g_j) \mid f'_i \otimes g'_j \in \frac{f' \otimes g'}{\biguplus}, f_i \otimes g_j \in \frac{f \otimes g}{\biguplus}, \right. \\
&\qquad \left. \operatorname{cod}(f_i) = \operatorname{dom}(f'_i), \operatorname{cod}(g_j) = \operatorname{dom}(g'_j) \right\} \\
&= \left\{ (f'_i \otimes g'_j) \circ (f_i \otimes g_j) \mid f'_i \circ f_i \in \frac{f' \circ f}{\biguplus}, g'_j \circ g_j \in \frac{g' \circ g_j}{\biguplus} \right\} \\
&= (\text{from the fact that all } f'_i, f_i, g'_j, g_j \text{ are simple arrows and (1)}) \\
&= \left\{ (f'_i \circ f_i) \otimes (g'_j \circ g_j) \mid f'_i \circ f_i \in \frac{f' \circ f}{\biguplus}, g'_j \circ g_j \in \frac{g' \circ g_j}{\biguplus} \right\} \\
&= \left\{ \left\{ \mathrm{id}_R \mid R \in \widetilde{f'_i \circ f_i} \bigcap \widetilde{g'_j \circ g_j} \right\} \mid f'_i \circ f_i \in \frac{f' \circ f}{\biguplus}, g'_j \circ g_j \in \frac{g' \circ g_j}{\biguplus} \right\} \\
&= (\text{by Definition 43}) \\
&= \frac{(f' \circ f) \otimes (g' \circ g)}{\biguplus}.
\end{aligned}$$

Consequently, $(f' \circ f) \otimes (g' \circ g) = (f' \otimes g') \circ (f \otimes g)$, and hence, the compositional property for functor $\otimes$ is satisfied.

Let us show the functorial property for the identity arrows, and consider two objects $A = \biguplus_{1 \le j \le m} A_j$ and $B = \biguplus_{1 \le i \le k} B_i$, with their identity arrows $\mathrm{id}_A = \biguplus_{1 \le j \le m} \mathrm{id}_{A_j}$ and $\mathrm{id}_B = \biguplus_{1 \le i \le k} \mathrm{id}_{B_i}$, such that $\widetilde{\mathrm{id}_A} = \biguplus_{1 \le j \le m} \widetilde{\mathrm{id}_{A_j}} = \biguplus_{1 \le j \le m} TA_j = TA$, $\widetilde{\mathrm{id}_B} = \biguplus_{1 \le i \le k} \widetilde{\mathrm{id}_{B_i}} = \biguplus_{1 \le i \le k} TB_i = TB$, with $\frac{\mathrm{id}_A}{\biguplus} = \{\mathrm{id}_{A_1}, \ldots, \mathrm{id}_{A_m}\}$ and $\frac{\mathrm{id}_B}{\biguplus} = \{\mathrm{id}_{B_1}, \ldots, \mathrm{id}_{B_k}\}$. Then

$$\begin{aligned}
\frac{\mathrm{id}_A \otimes \mathrm{id}_B}{\biguplus} &= \left\{ \{q_\perp\} \bigcup \left\{ \mathrm{id}_R \mid R \in \widetilde{\mathrm{id}_{A_j}} \bigcap \widetilde{\mathrm{id}_{B_i}} \right\} \neq \perp^1 \mid \mathrm{id}_{A_j} \in \frac{\mathrm{id}_A}{\biguplus}, \mathrm{id}_{B_i} \in \frac{\mathrm{id}_B}{\biguplus} \right\} \\
&= \left\{ \{q_\perp\} \bigcup \left\{ \mathrm{id}_R \mid R \in TA_j \bigcap TB_i \right\} \neq \perp^1 \mid 1 \le j \le m, 1 \le i \le k \right\} \\
&= \left\{ \mathrm{id}_{TA_j \cap TB_i} \mid TA_j \bigcap TB_i \neq \perp^0, 1 \le j \le m, 1 \le i \le k \right\} \\
&= \{ \mathrm{id}_{A_j \otimes B_i} \mid A_j \otimes B_i \neq \perp^0, 1 \le j \le m, 1 \le i \le k \} = \frac{\mathrm{id}_{A \otimes B}}{\biguplus}.
\end{aligned}$$

Hence, from Definition 44, $\mathrm{id}_A \otimes \mathrm{id}_B = \mathrm{id}_{A\otimes B} : A \otimes B \to A \otimes B$.

It is easy to verify that $\otimes$ is a monoidal bifunctor with natural isomorphic transformations (which generate an isomorphic arrow for each object in **DB**):

$$\begin{aligned}
&\alpha : (_ \otimes _) \otimes _ \longrightarrow _ \otimes (_ \otimes _), \quad \text{(associativity)}\\
&\beta : \Upsilon \otimes _ \longrightarrow I_{\mathrm{DB}}, \quad \text{(left identity)}\\
&\gamma : _ \otimes \Upsilon \longrightarrow I_{\mathrm{DB}}, \quad \text{(right identity)}
\end{aligned}$$

such that $A\otimes B \simeq B\otimes A$, $A\otimes\Upsilon = \Upsilon\otimes A \simeq A$ and $A\otimes\perp^0 \simeq \perp^0$. For any morphism $f : A \longrightarrow B$, from Proposition 23 we obtain $\widetilde{f} \preceq A \otimes B$. □

A tensor product $\otimes$ of the monoidal category **DB** *is not* unique in contrast with the Cartesian product (we can have for simple databases, $A\otimes B = C\otimes B$ such that $C = A \bigcup A_1 \supset A$ with $TA_1 \bigcap TB = \perp^0$).

Notice that each $A\otimes B$ is a closed object (intersection of two closed objects TA and TB), and that the information flux of any morphism from A to B is a closed object included in this maximal information flux (i. e., overlapping) between A and B. A matching of two completely disjoint simple databases is equal to the empty zero object $\perp^0$.

Proposition 37. *Each object A with the monomorphism $\mu_A : A \otimes A \longrightarrow A$ and the epimorphism $\eta_A : \Upsilon \twoheadrightarrow A$ compose a monoid in the monoidal category* $(\mathbf{DB}, \otimes, \Upsilon, \alpha, \beta, \gamma)$.

Proof. We have $A \otimes A \preceq A$, and hence, $\mu_A = \mathrm{in} : A \otimes A \longrightarrow A$ is a monomorphism (from Theorem 17 in Section 7.4). From the fact that $TA \subseteq T\Upsilon = \Upsilon$ and point 2 of Proposition 24, we obtain a monomorphism $\mathrm{in}_{TA} : TA \hookrightarrow \Upsilon$, and hence, a monomorphism (composition of two monic arrows) $in = \mathrm{in}_{TA} \circ \mathrm{is}_A : A \hookrightarrow \Upsilon$ with $\frac{in}{\biguplus} = \frac{\mathrm{id}_A}{\biguplus}$. Consequently, by duality, its opposite arrow $\eta_A = \mathrm{in}^{\mathrm{OP}} : \Upsilon \twoheadrightarrow A$ is an epimorphism such that $\widetilde{\eta_A} = \widetilde{\mathrm{in}^{\mathrm{OP}}} = \widetilde{in} \simeq \widetilde{\biguplus \frac{in}{\biguplus}} = \widetilde{\biguplus \frac{\mathrm{id}_A}{\biguplus}} = \widetilde{\mathrm{id}_A} = TA$. It is easy to verify that $\mu_A \circ (\mu_A \otimes \mathrm{id}_A) \circ \alpha_{A,A,A} = \mu_A \circ (\mathrm{id}_A \otimes \mu_A)$ is valid, and $\beta_A = \mu_A \circ (\eta_A \otimes \mathrm{id}_A)$, $\gamma_A = \mu_A \circ (\mathrm{id}_A \otimes \eta_A)$. □

We generalize the merging operator for objects for the set $S = \{A_1, \ldots, A_n\}$ of more than two databases by $\otimes S = A_1 \otimes \cdots \otimes A_n$.

7.4.2 Merging operator

Merging of an instance-database A with another instance-database is a unary algebraic operation, similar to the binary operation Data Federation of two database instances (denoted by $\frac{\bigcup}{rn}$ in Section 7.2.1, i. e., the union $\bigcup$ with the renaming (rn) of the relational symbols in $\mathcal{A}$ and $\mathcal{B}$ that have originally the same names, by unifying these two databases, that are in two different DB machines, for example, under the common DBMS. Consequently, given a connection $\mathcal{A} \bigoplus \mathcal{B}$ of two schemas and an R-algebra α (which is their model), $A \frac{\bigcup}{rn} B = \alpha^* (\mathcal{A} \bigoplus \mathcal{B}))$.

In fact, the merging of two databases is defined as the database obtained by applying the power-view operator T to data federation of these two databases. That is, the data federation of two databases is isomorphic to the database obtained by their merging, from the behavioral point of view. Any view which can be obtained from the data federation (union with renaming) of two databases can also be obtained from the merging of these two databases, and vice versa.

In what follows, similar to the matching tensor products which, for any two given databases, return a closed object, also the merging operator will return a closed object. As we will see, these two operators will provide the meet and joint operators of the complete algebraic database lattice, where $\perp^0$ and Υ are the bottom and top elements (i. e., the instance-databases), respectively.

Theorem 19. *For any fixed simple database $A \in \mathrm{Ob}_{\mathrm{DB}}$, we define the parameterized "merging with A" operator as an endofunctor $A \oplus _ : \mathbf{DB} \longrightarrow \mathbf{DB}$, as follows:*

1. *For any object $B = \biguplus_{1\leq i\leq k} B_i$, $k \geq 1$, the object $A \oplus B$ is a merging of A, defined by:*

$$A \oplus B \equiv \oplus(A,B) \triangleq \begin{cases} T(A \underset{rn}{\bigcup} B) = T(A \bigcup B), & \text{if } k = 1, B = B_1; \\ \biguplus_{1\leq i\leq k}(A \oplus B_i), & \text{otherwise.} \end{cases}$$

2. *For any $f : B \to \biguplus_{1\leq j\leq m} C_j$, $m \geq 1$, the arrow $A \oplus (f) : A \oplus B \to A \oplus C$ is defined by the canonical representation (Lemma 2) of*

$$\frac{A \oplus (f)}{\biguplus} \triangleq \left\{ \left(\mathrm{id}_A \underset{rn}{\bigcup} f_{ij} \right) : A \oplus B_i \to A \oplus C_j \mid f_{ij} : B_i \to C_j \in \frac{f}{\biguplus}, \left(\mathrm{id}_A \underset{rn}{\bigcup} f_{ij} \right) \neq \perp^1 \right\}.$$

Proof. Let us show that for a given **DB** simple arrow $f : B \to C$, and hence, from Theorem 10, obtained from a schema mapping $\mathcal{M}_{BC} : \mathcal{B} \to \mathcal{C}$ and an R-algebra α, such that $f = \alpha^*(\mathrm{MakeOperads}(\mathcal{M}_{BC}))$ with the simple databases $B = \alpha^*(\mathcal{B})$ and $C = \alpha^*(\mathcal{C})$, there is also an arrow $\mathrm{id}_A \underset{rn}{\bigcup} f$ in **DB**. In fact, let us consider the identity schema mapping $\mathcal{M}_{AA} : \mathcal{A} \to \mathcal{A}$ and the sketch's mapping $\mathbf{M} = (\mathrm{MakeOperads}(\mathcal{M}_{AA}) \underset{rn}{\bigcup} \mathrm{MakeOperads}(\mathcal{M}_{BC})) : \mathcal{A} \bigoplus \mathcal{B} \to \mathcal{A} \bigoplus \mathcal{C}$. Then

$$\begin{aligned} &\alpha^*\left(\mathrm{MakeOperads}(\mathcal{M}_{AA}) \underset{rn}{\bigcup} \mathrm{MakeOperads}(\mathcal{M}_{BC}) \right) \\ &= \alpha^*\left(\mathrm{MakeOperads}(\mathcal{M}_{AA})\right) \underset{rn}{\bigcup} \alpha^{(}\mathrm{MakeOperads}(\mathcal{M}_{BC})) \\ &= \mathrm{id}_A \underset{rn}{\bigcup} f : A \underset{rn}{\bigcup} B \to A \underset{rn}{\bigcup} B \\ &= A \oplus (f) : A \underset{rn}{\bigcup} B \to A \underset{rn}{\bigcup} B. \end{aligned}$$

From the fact that $A \oplus B \simeq A \underset{rn}{\bigcup} B$ and $A \oplus C \simeq A \underset{rn}{\bigcup} C$, we obtain the **DB** arrow $\mathrm{id}_A \underset{rn}{\bigcup} f : A \oplus B \to A \oplus C$.

It is easy to verify that for any two simple databases A and B the database $A \underset{rn}{\bigcup} B$ is different from $A \bigcup B$ only in the fact that by renaming we can have the copies of the same relations. Thus, the set of views (obtained by SPRJU queries) of both databases are equal. Consequently, $T(A \underset{rn}{\bigcup} B) = T(A \bigcup B) = T(TA \bigcup TB) = T(TA \underset{rn}{\bigcup} TB)$, i. e., $A \oplus B = TA \oplus TB$, and $A \underset{rn}{\bigcup} B \simeq A \oplus B$.

Now we can verify that $A \oplus _$ is an endofunctor. In fact, for any object $B = \biguplus_{1 \leq i \leq k} B_i$, $k \geq 1$ and its identity arrow $\mathrm{id}_B = \biguplus_{1 \leq i \leq k} (\mathrm{id}_{B_i} : B_i \to B_i)$,

$$\begin{aligned} \frac{A \oplus (\mathrm{id}_B)}{\biguplus} &= \left\{ \left(\mathrm{id}_A \underset{rn}{\bigcup} \mathrm{id}_{B_i} \right) : A \oplus B_i \to A \oplus B_i \mid \mathrm{id}_{B_i} : B_i \to B_i \in \frac{\mathrm{id}_B}{\biguplus} \right\} \\ &= \left\{ A \oplus (\mathrm{id}_{B_i}) : A \oplus B_i \to A \oplus B_i \mid \mathrm{id}_{B_i} : B_i \to B_i \in \frac{\mathrm{id}_B}{\biguplus} \right\}. \end{aligned}$$

Thus, for each simple point-to-point arrow $A \oplus (\mathrm{id}_{B_i}) : A \oplus B_i \to A \oplus B_i$, its flux is $\widetilde{A \oplus (\mathrm{id}_{B_i})} = \widetilde{(\mathrm{id}_A \underset{rn}{\bigcup} \mathrm{id}_{B_i})} = \widetilde{\mathrm{id}_{A \underset{rn}{\bigcup} B_i}} = T(A \underset{rn}{\bigcup} B_i) = A \oplus B_i$. Consequently, for the identity arrows is valid the functorial property, $A \oplus (\mathrm{id}_{B_i}) = \mathrm{id}_{A \oplus B_i} : A \oplus B_i \to A \oplus B_i$.

From the fact that for any simple object (i. e., a database) B, $A \subseteq A \oplus B$, each point-to-point arrow, resulting by application of this endofunctor, contains a subarrow id_A. Thus, based on composition of operads in point 4 of Definition 63 (Section A.3.3), given two arrows $f : B \to C$ and $g : C \to D$, we have the compositional endofunctors property:

$$\begin{aligned} \frac{A \oplus (g) \circ A \oplus (f)}{\biguplus} &= \left\{ \left(\mathrm{id}_A \underset{rn}{\bigcup} g_{jl} \right) \circ \left(\mathrm{id}_A \underset{rn}{\bigcup} f_{ij} \right) \mid g_{jl} \circ f_{ij} \in \frac{g \circ f}{\biguplus} \right\} \\ &= \left\{ \mathrm{id}_A \underset{rn}{\bigcup} (g_{jl} \circ f_{ij}) \mid g_{jl} \circ f_{ij} \in \frac{g \circ f}{\biguplus} \right\} = \frac{A \oplus (g \circ f)}{\biguplus}. \end{aligned}$$

Consequently, $A \oplus (g) \circ A \oplus (f) = A \oplus (g \circ f)$.

Moreover, $A \oplus B = B \oplus A$ (if B is a simple database as well), $A \oplus \Upsilon = \Upsilon$, $A \oplus \bot^0 \simeq A$, $A \oplus TA = TA$ and $A \oplus A \simeq A$. □

"The merging with a complex-object A," when $A = \biguplus_{1 \leq j \leq m} A_j$, $m \geq 2$, can be obtained from the object-component of the functor defined in the theorem above, will be considered in more details in Section 7.4.4, by obtaining the symmetric operator '$\widehat{\oplus}$', such that for any two complex objects A and B the isomorphism $A \widehat{\oplus} B \simeq B \widehat{\oplus} A$ is valid.

Matching '$\otimes$' and (generalized) merging '$\widehat{\oplus}$' operators are dual operators in the category **DB**: in fact, they are also two dual lattice operators (meet and join, respectively) w. r. t. the database ordering $\preceq$, as we will show in Section 7.4.4.

Remark. Notice that $A \underset{rn}{\bigcup} B \simeq A \oplus B$, i. e., from the behavioral point of view, the data federation is equivalent (isomorphic) to the data merging. That is, for any query over the data federation $A \underset{rn}{\bigcup} B$, which returns a view R, there exists a query over the data merging $A \oplus B$, which returns with the same view R and vice versa.

We generalize the merging operator for objects for the set $S = \{A_1, \ldots, A_n\}$ of more than two simple databases by $\oplus S = T(\bigcup S) = T(A_1 \bigcup \cdots \bigcup A_n)$.

7.4.3 Universal algebra considerations

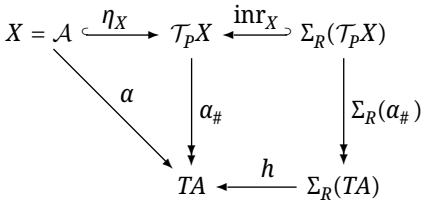
In what follows, we will consider the signature Σ_R of the SPRJU Codd's relational algebra, which is used to define the views in the SOtgd implications for the schema database mappings. In order to explore the universal algebra properties of the category **DB** [64], where a morphism is not a function but a nonempty *set* of functions (this fact significatively complicates a definition of the mappings from **DB**-morphisms into the homomorphisms of the category of Σ_R-algebras), we will use an, equivalent to **DB**, a "sets-like" category $\mathbf{DB}_{\mathrm{sk}}$, such that its arrows can be seen as total functions.

Proposition 38. *Let us denote the full skeletal subcategory of* **DB** *composed of closed objects only by* $\mathbf{DB}_{\mathrm{sk}}$.

Such a category is equivalent to the category **DB**, *i. e., there exist an adjunction of a surjective functor* $T_{\mathrm{sk}} : \mathbf{DB} \longrightarrow \mathbf{DB}_{\mathrm{sk}}$ *and an inclusion functor* $\mathrm{In}_{\mathrm{sk}} : \mathbf{DB}_{\mathrm{sk}} \longrightarrow \mathbf{DB}$ *such that* $T_{\mathrm{sk}}\mathrm{In}_{\mathrm{sk}} = \mathrm{Id}_{DB_{\mathrm{sk}}}$ *and* $\mathrm{In}_{\mathrm{sk}}T_{\mathrm{sk}} \simeq \mathrm{Id}_{\mathrm{DB}}$.

Proof. Let us define $T^0_{\mathrm{sk}} = T^0$ and $T^1_{\mathrm{sk}} = T^1$, while $\mathrm{In}^0_{\mathrm{sk}}$ and $\mathrm{In}^1_{\mathrm{sk}}$ are two identity functions. It is easy to verify that these two categories are equivalent. In fact, there exists an adjunction $(T_{\mathrm{sk}}, \mathrm{In}_{\mathrm{sk}}, \eta_{\mathrm{sk}}, \varepsilon_{\mathrm{sk}}) : \mathbf{DB} \longrightarrow \mathbf{DB}_{\mathrm{sk}}$ based on the bijection $\sigma : \mathbf{DB}_{\mathrm{sk}}(T_{\mathrm{sk}}A, B) \to \mathbf{DB}(A, \mathrm{In}_{\mathrm{sk}}B)$, which is natural in simple objects $A \in \mathbf{DB}$ and $B \in \mathbf{DB}_{\mathrm{sk}}$ (thus, B is closed, i. e., $B = TB$). In fact, for each $f \in \mathbf{DB}_{\mathrm{sk}}(T_{\mathrm{sk}}A, B)$, which is an arrow $f : TA \to B$ (from $T_{\mathrm{sk}}A = TA$), we define $\sigma(f) = f \circ \mathrm{is}_A : A \to B$; conversely, for any $g \in \mathbf{DB}(A, \mathrm{In}_{\mathrm{sk}}B)$, which is an arrow $g : A \to B$ (from $\mathrm{In}_{\mathrm{sk}}B = B$), we define $\sigma^{-1}(g) = g \circ \mathrm{is}^{-1}_A$. Thus, $\sigma^{-1}(\sigma(f)) = \sigma(f) \circ \mathrm{is}^{-1}_A = f \circ \mathrm{is}_A \circ \mathrm{is}^{-1}_A = f \circ \mathrm{id}_{TA} = f$, and $\sigma(\sigma^{-1}(g)) = \sigma(g \circ \mathrm{is}^{-1}_A) = g \circ \mathrm{is}^{-1}_A \circ \mathrm{is}_A = g \circ \mathrm{id}_A = g$. □

In a given inductive definition, one defines a value of a function on all (algebraic) constructors (i. e., relational operators), in our example the endofunctor T, by the initial relational $(X \biguplus \Sigma_R)$-algebra semantics with the syntax monad $(\mathcal{T}_P, \eta, \mu)$. It is provided for a given assignment (R-algebra) $\alpha : X \to TA$, where X is a set of relational variables (symbols) in a database schema $\mathcal{A}$, such that $A = \alpha^*(\mathcal{A})$. This inductive definition is expressed by the following commutative diagram in Set:



As we see, the *induction* for each schema $\mathcal{A}$ and R-algebra α, which is an interpretation of $\mathcal{A}$, defines the *closed objects* in **DB**.

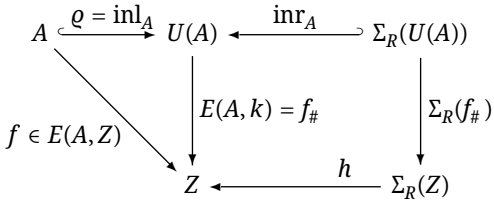
The following is based on the fundamental results of the universal algebra [102].

Based on the Cod's SPRJU relational algebra, Σ_R is a finitary signature (in the usual algebraic sense: a collection Σ_R of function *symbols* with a function $\mathrm{ar} : \Sigma_R \longrightarrow \mathcal{N}$ giving the finite arity of each function symbol) for a single-sorted (sort of relations) relational algebra. Here, we will recall some universal algebra concepts from Section A.1:

We can speak of Σ_R-equations E and their satisfaction in a Σ_R-algebra, obtaining the notion of a (Σ_R, E)-algebra theory. In a special case when E is empty, we obtain a pure syntax version of Universal algebra, where **K** is a category of all Σ_R-algebras and the quotient-term algebras are simply term algebras $\mathcal{T}_P X$ for a given set of relational variables X.

An *algebra* for the algebraic theory (or type) (Σ_R, E) is given by a set X, called the *carrier* of the algebra, together with interpretations for each of the function symbols in Σ_R. A function symbol $o_i \in \Sigma_R$ of arity k must be interpreted by a function $\widehat{o}_i : X^k \longrightarrow X$. Given this, a term containing n distinct variables gives rise to a function $X^n \longrightarrow X$ defined by induction on the structure of the term. An algebra must also satisfy the equations given in E in the sense that equal *terms* give rise to identical functions (with obvious adjustments where the equated terms do not contain exactly the same variables). A *homomorphism* of algebras from an algebra X to an algebra Y is given by a function $g : X \longrightarrow Y$, which commutes with operations of the algebra $g(\widehat{o}_i(x_1, \ldots, x_k)) = \widehat{o}_i(g(x_1), \ldots, g(x_k))$.

This generates a variety category **K** of all relational algebras. Consequently, there is a bifunctor $E : \mathbf{DB}_{\mathrm{sk}}^{\mathrm{OP}} \times \mathbf{K} \longrightarrow \mathbf{Set}$ such that for any *simple* database instance A in $\mathbf{DB}_{\mathrm{sk}}$ there exists the functor $E(A, _) : \mathbf{K} \longrightarrow \mathbf{Set}$ with an universal element $(U(A), \varrho)$, where $\varrho \in E(A, U(A))$, i. e., $\varrho : A \longrightarrow U(A)$, is an inclusion function. $U(A)$ is a free algebra over A (i. e., the quotient-term algebra generated by a carrier database instance A) such that for any function $f \in E(A, Z), f : A \to Z$, there is a unique homomorphism k (in **K**) from the free algebra $U(A)$ into an algebra Z, with $f = E(A, k) \circ \varrho$, represented by the following initial algebra semantics diagram in **Set**:



where $k : (U(A), \mathrm{inr}_A) \to (Z, h)$ is this unique Σ_R-algebra homomorphism corresponding to the unique homomorphism k from the initial $(A \biguplus \Sigma_R)$-algebra $(U(A), [\mathrm{inl}_A, \mathrm{inr}_A])$ into the algebra $(Z, [f, k])$, and $E(A, k) = f_\#$ is a function equal to k but without the structural properties of this homomorphism (to commute with operations of the algebra).

Consequently, $U(A)$ is a quotient-term algebra, where carrier is a set of equivalence classes of closed terms of a well-defined formulae of a relational algebra, "constructed"

by Σ_R-constructors (relational operators in SPJRU algebra: select, project, join and union) and elements (relations) of a database instance A.

From the so-called "parameter-theorem," we obtain that there exist:

– A unique universal functor $U : \mathbf{DB}_{\mathrm{sk}} \longrightarrow \mathbf{K}$ such that for any simple database instance A in $\mathbf{DB}_{\mathrm{sk}}$ it returns the free Σ_R-algebra $U(A)$. It is a quotient-term algebra where a carrier is a set $U(A)$ of the equivalence classes $[t_i]$ of closed terms $t_i \in \mathcal{T}_P A$ of well-defined formulae of a relational algebra, "constructed" by Σ_R-constructors (i. e., the relational operators $o_i \in \Sigma_R$ in SPJRU algebra: select, project, join and union) and relational symbols of a database instance A. That is, $U(A) = \{[t_i] \mid t_i \in \mathcal{T}_P A\}$. An alternative for $U(A)$ is given by considering A as a set of variables rather than a set of constants. In either case, each k-ary operation $o_i \in \Sigma_R$, with the function $\widehat{o_i} : U(A)^k \to U(A)$ (i. e., a component of $\mathrm{inr}_A : \Sigma_R(U(A)) \to U(A)$ where $\Sigma_R(U(A)) = \biguplus_{o_i \in \Sigma_R} U(A)^{\mathrm{ar}(o_i)}$, so that $\mathrm{inr}_A = \biguplus_{o_i \in \Sigma_R} \widehat{o_i}$ is the interpretation of all operations in the signature Σ_R), and hence, is interpreted syntactically $\widehat{o_i}([t_1], \ldots, [t_k]) = [o_i(t_1, \ldots, t_k)]$, where as usual, the square brackets denote the equivalence classes for terms t_i.

This universal functor U, for any simple morphism $f : A \longrightarrow B$ in $\mathbf{DB}_{\mathrm{sk}}$ returns the homomorphism $f_H = U^1(f)$ from the Σ_R-algebra $U(A)$ into the Σ_R-algebra $U(B)$ such that for any term $o_i([t_1], \ldots, [t_n]) \in U(A)$, $o_i \in \Sigma_R$, we obtain $f_H(o_i([t_1], \ldots, [t_n])) = o_i(f_H([t_1]), \ldots, f_H([t_n])) \in U(B)$.

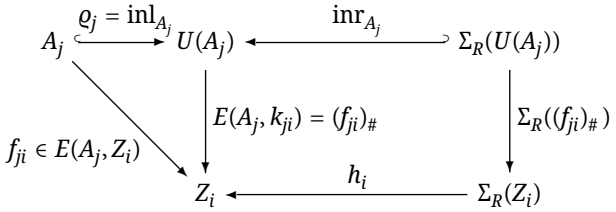
– Its adjoint forgetful functor $F : \mathbf{K} \longrightarrow \mathbf{DB}_{\mathrm{sk}}$ such that, for any free algebra $U(A)$ in $\mathbf{K}$, the object $F(U(A))$ in $\mathbf{DB}_{\mathrm{sk}}$ is equal to its carrier-set A (each term $o_i([t_1], \ldots, [t]_n) \in U(A)$ is evaluated into a relation that is an element of this closed object A) and for each arrow $U^1(f)$, $F^1(U^1(f)) = f$, i. e., $FU = \mathrm{Id}_{DB_{\mathrm{sk}}}$ and $UF = \mathrm{Id}_K$.

In this universal algebra framework, we preferred to use the skeletal category $\mathbf{DB}_{\mathrm{sk}}$ (which will be used for the algebraic considerations in what follows), instead of the complete category **DB**, equivalent to it.

It is immediate from the universal property that the mapping $A \mapsto U(A)$ extends to the endofunctor $F \circ U : \mathbf{DB}_{\mathrm{sk}} \longrightarrow \mathbf{DB}_{\mathrm{sk}}$. This functor carries a *monad structure* $(F \circ U, \eta, \mu)$ with $F \circ U$ an equivalent version of T but for this skeletal database category $\mathbf{DB}_{\mathrm{sk}}$ (while for the equivalent version for **DB** we obtain the monad T). The natural transformation η is given by the obvious "inclusion" of A into $(F \circ U)(A) = F(U(A)) : R \longrightarrow [R]$ (each view (relation) R in a closed object A is an equivalence class of all algebra terms, which produce this view). The natural transformation η is the unit of this adjunction of U and F, and hence, it corresponds to an injective function $\varrho : A \longrightarrow U(A)$ in **Set**, given previously. The interpretation of μ is almost equally simple. An element of $(F \circ U)^2(A)$ is an equivalence class of terms built up from elements of $(F \circ U)(A)$, so that instead of $t(x_1, \ldots, x_k)$, a typical element of $(F \circ U)^2(A)$ is given by the equivalence class of a term $t([t_1], \ldots, [t_k])$. The transformation μ is defined by "flattening-terms" map $[t([t_1], \ldots, [t_k])] \mapsto [t(t_1, \ldots, t_k)]$. This make sense because a substitution of provably equal expressions into the same term results in provably equal terms.

Remark. Consider a generalization to complex objects $A = \biguplus_{1\le j\le m} A_j$, $m \ge 2$:

A bifunctor $E : \mathbf{DB}_{\mathrm{sk}}^{\mathrm{OP}} \times \mathbf{K} \longrightarrow \mathbf{Set}$ for any *complex* database instance A in $\mathbf{DB}_{\mathrm{sk}}$ generates a functor $E(A, _) : \mathbf{K} \longrightarrow \mathbf{Set}$ with a universal element $(U(A), \varrho)$, where $U(A) = \biguplus_{1\le j\le m} U(A_j)$ and $\varrho = (\biguplus_{1\le j\le m} \varrho_j) \in E(A, U(A))$, i. e., $\varrho : \biguplus_{1\le j\le m} A_j \longrightarrow \biguplus_{1\le j\le m} U(A_j)$ is an inclusion function and $U(A)$ is a free algebra over A (i. e., a quotient-term algebra generated by a carrier database instance A). It is a disjoint union of the free algebras $U(A_j)$ over A_j, $j = 1, \ldots, m$, such that for any function $f \in E(A, Z), f : A \to Z$, with a set of point-to-point functions $f_{ji} : A_j \to Z_i \in \overset{f}{\biguplus}$ for a complex object $Z = \biguplus_{1\le j\le k} Z_i$, $k \ge 1$, for each f_{ji} there is a unique homomorphism k_{ji} from the free algebra $U(A_j)$ into an algebra Z_i, with $f_{ji} = E(A_j, k) \circ \varrho_j$, represented by the following initial algebra semantics diagram in **Set**:



The set of such unique point-to point homomorphisms $k_{ji} : (U(A_j), \mathrm{inr}_{A_j}) \to (Z_i, h_i)$, for each $f_{ji} \in \overset{f}{\biguplus}$ define $\overset{k}{\biguplus} = \{k_{ji} \mid f_{ji} \in \overset{f}{\biguplus}\}$, i. e., a unique Σ_R-algebra homomorphism $k : (U(A), \mathrm{inr}_A) \to (Z, h)$, which corresponds to the unique homomorphism k from the initial $(A \biguplus \Sigma_R)$-algebra $(U(A), [\mathrm{inl}_A, \mathrm{inr}_A])$ into the algebra $(Z, [f, k])$.

Thus, $E(A, k) = f_\#$ is a function equal to k but without the structural properties of the homomorphism (to commute with operations of the algebra). □

Let us define now the faithful forgetful functor from **DB** into **Set** category.

Theorem 20 ([5]). *There exists a faithful forgetful functor $F_{\mathrm{sk}} = (F^0_{\mathrm{sk}}, F^1_{\mathrm{sk}}) : \mathbf{DB}_{\mathrm{sk}} \longrightarrow \mathbf{Set}$, such that its object-component F^0_{sk} is equal to powerset operator $\mathcal{P}$, while the arrow-component F^1_{sk}, for a given morphism $f : A \to B$, where $A = \biguplus_{1\le j\le m} A_j$, $m \ge 1$, and $B = \biguplus_{1\le i\le k} B_i$, $k \ge 1$, is defined as follows: for any set $S \in F^0_{\mathrm{sk}}(A) = \mathcal{P}(\biguplus_{1\le j\le m} A_j)$,*

$$F^1_{\mathrm{sk}}(f)(S) \triangleq \begin{cases} \{R \mid R \in S \bigcap \widetilde{f_l}, (f_l : A \to B) \in \overset{f}{\biguplus}\}, & \text{if } m = k = 1; \\ \{R \mid (j, R) \in S \text{ and } R \in \widetilde{f_{j1}}, (f_{j1} : A_j \to B_1) \in \overset{f}{\biguplus}\}, & \text{if } m \ge 2,\ k = 1; \\ \{(i, R) \mid R \in S \bigcap \widetilde{f_{1i}}, (f_{1i} : A_1 \to B_i) \in \overset{f}{\biguplus}\}, & \text{if } m = 1,\ k \ge 2; \\ \{(i, R) \mid (j, R) \in S, R \in \widetilde{f_{ji}}, (f_{ji} : A_j \to B_i) \in \overset{f}{\biguplus}\}, & \text{otherwise.} \end{cases}$$

We define the faithful forgetful functor $F_{\mathrm{DB}} = F_{\mathrm{sk}} \circ T : \mathbf{DB} \longrightarrow \mathbf{Set}$ as well.

*Consequently, $\mathbf{DB}_{\mathrm{sk}}$ and **DB** are concrete categories.*

7.4.4 Algebraic database lattice

This section is an extension of the Section 7.4 dedicated to the partial ordering '$\preceq$' for databases with an introduction of the PO subcategory $\mathbf{DB}_I$.

We recall that the symbol '=' is used to express the fact that both sides name the same objects, whereas '$\approx$' is used to build equations, which may or may not be true for particular elements. Note that in the case of equations between arrows in **DB** category (i. e., the commutative diagrams) we use the symbol '=' instead of '$\approx$', and '=' is used for set equality relation as well.

We recall (see [102]) the fact that a complete lattice $\mathcal{C}$ is a poset P such that for any subset S both $\inf(S)$ (greatest lower bound) and $\sup(S)$ (least upper bound) exist in P (with $0 = \inf(P)$, $1 = \sup(P)$): for any $A, B, C \in \mathcal{C}$, and binary operations "join" $\bigvee$ and "meet" $\bigwedge$ (in the case of the **Set** category, these operators are the set-union and set-intersection, respectively, while for the **DB** category we will show that they are merging and matching operators, respectively), the following identities are satisfied (see the first 5 equations for complete lattices in Section A.1 in the Appendix)

1) $A \bigvee A \approx A, A \bigwedge A \approx A$; (idempotency)
2) $A \bigvee B \approx B \bigvee A, A \bigwedge B \approx B \bigwedge A$; (commutativity)
3) $A \bigvee (B \bigvee C) \approx (A \bigvee B) \bigvee A, A \bigwedge (B \bigwedge C) \approx (A \bigwedge B) \bigwedge C$; (associativity)
4) $A \vee 0 \approx A, A \wedge 1 \approx A$
5) $A \approx A \bigvee (A \bigwedge B), A \approx A \bigwedge (A \bigvee B)$; (absorption).

Let us denote the restriction of the closure endofunctor $T : \mathbf{DB} \longrightarrow \mathbf{DB}$ to PO category $\mathbf{DB}_I$ by $T_I : \mathbf{DB}_I \longrightarrow \mathbf{DB}_I$. We have seen in Section 7.4 that $\mathbf{DB}_I$ is a PO category where each arrow in $: A \hookrightarrow B$ is a monomorphism with $A \preceq B$ (as defined by Theorem 17 in Section 7.4). We recall that, by Definition 48 and Corollary 22 in Section 7.2.3, for the behavioral equivalence '$\approx$',

$$A \approx B \quad \text{iff} \quad (A \preceq B \text{ and } B \preceq A), \text{ and } A \simeq B \text{ implies } A \approx B.$$

Thus, we obtained a partial order $(\mathrm{Ob}_{\mathrm{DB}}, \preceq)$. Let us show that it is a lattice ordered set, i. e., that every pair of objects $A, B \in \mathrm{Ob}_{\mathrm{DB}}$ has the least-upper-bound (sup) and the greatest-lower-bound (inf).

First of all, if we consider the subset of only simple objects then $A \preceq B$ is equal to $TA \subseteq TB$, $A \otimes B = TA \bigcap TB$ and $A \oplus B = T(A \bigcup B) = T(TA \bigcup TB)$. Hence, the matching and merging operators correspond to the meet and join lattice operators. For the complex objects, we can continue to use the matching operator as meet operation, but we have to generalize the "merging with A" to complex objects as well, in the way that it can be used as a join operation.

Definition 53. Let $A = \biguplus_{1 \leq j \leq m} A_j$, $m \geq 1$, and $B = \biguplus_{1 \leq i \leq k} B_i$, $k \geq 1$, be two complex objects, then we define the following sets:

$$S_A = \{A_j \mid \exists B_i.A_j \succeq B_i, 1 \leq j \leq m \neq 1, 1 \leq i \leq k \neq 1\}.$$
$$S_B = \{B_i \mid \exists A_j.A_j \preceq B_i, 1 \leq j \leq m \neq 1, 1 \leq i \leq k \neq 1\}.$$
$$S_{AB} = \{A_j \oplus B_i \mid A_j \notin S_A, B_i \notin S_B, 1 \leq j \leq m, 1 \leq i \leq k\}.$$

We define the generalized merging operator '$\widehat{\oplus}$' as follows:

$$A\widehat{\oplus}B \equiv \widehat{\oplus}(A,B) \triangleq \begin{cases} A_1 \oplus B_1 = T(A_1 \bigcup B_1) \in S_{AB}, & \text{if } m = k = 1; \\ \biguplus(S_A \bigcup S_B \bigcup S_{AB}), & \text{otherwise.} \end{cases}$$

We recall (see the remark after Definition 42) that for a given set of sets $S = \{S_1, \ldots, S_n\}$, $\biguplus S$ represents the disjoint union of sets in S by taking *any ordering* of them (because they are equal up to isomorphism, i. e., $\biguplus S \simeq (S_1 \uplus S_2 \uplus \cdots \uplus S_n)$). We have the following interesting cases:

1. If $S_A = A$ then $S_{AB} = \emptyset$ (empty set), and hence, $A \preceq B$;
2. If $S_B = B$ then $S_{AB} = \emptyset$, and hence, $A \succeq B$;
3. If $S_A = A$ and $S_B = B$ then $S_{AB} = \emptyset$, and hence, $A \approx B$;
4. If $S_A = S_B = \emptyset$, then S_{AB} is the disjoint union of all possible mergings of the simple objects (i. e., databases) in A and B, with $A \not\preceq B$ and $B \not\preceq A$.

We have the following important properties for this generalization of the "merging with A" operator '$\widehat{\oplus}$'.

Lemma 9. *The following properties for any two objects A and B in* **DB** *are valid:*

1. *(Commutativity) $A\widehat{\oplus}B \simeq B\widehat{\oplus}A$; thus, $A\widehat{\oplus}B \approx B\widehat{\oplus}A$;*
2. *(Generalized merging) If A is a simple object, then $A\widehat{\oplus}B \simeq A \oplus B$. Thus, $A\widehat{\oplus}B \approx A \oplus B$;*
3. *$A\widehat{\oplus}B \succeq A$ and $A\widehat{\oplus}B \succeq B$;*
4. *If $A \preceq B$, then $A\widehat{\oplus}B \approx B$.*

Proof. Claim 1. From $A\widehat{\oplus}B = \biguplus(S_A \bigcup S_B \bigcup S_{AB})$, we obtain the isomorphism (from the commutativity of $\biguplus$), $A\widehat{\oplus}B \simeq B\widehat{\oplus}A$, and hence, $A\widehat{\oplus}B \approx B\widehat{\oplus}A$ (from Corollary 22 any two isomorphic objects are behaviorally equivalent), which represents the commutativity property of the generalized merging operator.

Claim 2. It is a generalization of the "merging with A" operator in Theorem 19, from the fact that when $m = 1$ (i. e., A is a simple object), we obtain $S_A = S_B = \emptyset$, and hence, $A\widehat{\oplus}B = \biguplus S_{AB} = \biguplus\{A_1 \oplus B_i \mid 1 \leq i \leq k\} \simeq \biguplus_{1 \leq i \leq k} A_1 \oplus B_i =$ (from Theorem 19) $= A \oplus B$. Thus, $A\widehat{\oplus}B \approx A \oplus B$.

Claim 3. For each $1 \leq j \leq m$, the simple object A_j is in S_A or in S_{AB} in the form $A_j \otimes B_i \supseteq TA_j$, and hence, $A\widehat{\oplus}B \succeq A$. Analogously, for each $1 \leq i \leq k$, the simple object B_i is in S_B or in S_{AB} in form $A_j \otimes B_i \supseteq TB_i$, so that $A\widehat{\oplus}B \succeq B$.

Claim 4. From the fact that $A\widehat{\oplus}B \succeq B$, let us show that if $A \preceq B$ (with a mapping $\sigma : \{1, \ldots, m\} \to \{1, \ldots, k\}$ such that $\forall_{1 \leq j \leq m}.TA_j \subseteq TB_{\sigma(j)}$) then $A\widehat{\oplus}B \preceq B$ as well. In fact, if

$A \preceq B$ then $S_B = B$ and $S_{AB} = \emptyset$. Thus, the elements in the disjoint union of $A\hat{\oplus}B$ are all B_i, with $TB_i \subseteq TB_i, 1 \leq i \leq k$, or some A_j, so that $TA_j \subseteq TB_{\sigma(j)}$, and hence, $A\hat{\oplus}B \preceq B$. □

We are now ready to introduce the following database lattice L_{DB}.

Proposition 39. *The set* $\mathrm{Ob}_{\mathrm{DB}}$ *of all database instances (objects) of* **DB***, both with the generalized merging and matching tensor products* $\hat{\oplus}$ *and* $\otimes$ *(read "join" and "meet," respectively) is a lattice with partial ordering '*$\preceq$*' (introduced by point 3 in Definition* 40*, Section* 7.1*).*

This lattice $L_{\mathrm{DB}} = (\mathrm{Ob}_{\mathrm{DB}}, \preceq, \otimes, \hat{\oplus})$ *is a complete lattice with the top and bottom objects* Υ *and* $\perp^0$*, respectively.*

Proof. First of all, if $A \preceq B$ with a mapping $\sigma : \{1, \ldots, m\} \to \{1, \ldots, k\}$ such that $\forall_{1 \leq j \leq m}.TA_j \subseteq TB_{\sigma(j)}$, we obtain, from points 3 and 4 of Lemma 9, that the generalized merging '$\hat{\oplus}$' is equal to the join operator of this lattice w. r. t. the ordering '$\preceq$'. Let us show that the matching operator $\otimes$ is the meet operator of this lattice. In fact, from Theorem 18, $A \otimes B \preceq A$ and $A \otimes B \preceq B$ (for each disjoint component $A_j \otimes B_i$ in $A \otimes B$, $A_j \otimes B_i = TA_j \bigcap TB_i \subseteq TA_j$ and $A_j \otimes B_i = TA_j \bigcap TB_i \subseteq TB_i$). If $A \preceq B$ with a mapping $\sigma : \{1, \ldots, m\} \to \{1, \ldots, k\}$ such that $\forall_{1 \leq j \leq m}.TA_j \subseteq TB_{\sigma(j)}$, then $A \preceq A \otimes B$ as well (we define a mapping $\sigma' : \{1, \ldots, m\} \to \{(1,1), \ldots, (1,k), \ldots, (m,1), \ldots (m,k)\}$ such that $\sigma'(j) = (j, \sigma(j)), 1 \leq j \leq m$, with $TA_j \subseteq TA_j \otimes TB_{\sigma(j)} = TA_j \bigcap TB_{\sigma(j)} = TA_j$). Thus, if $A \preceq B$ then $A \otimes B \approx A$.

Hence, $\inf(A, B) = A \otimes B$ and $\sup(A, B) = A\hat{\oplus}B$.

Consequently, the equivalence in this lattice is equal to the strong behavioral equivalence of **DB** category (from Corollary 22 in Section 7.2.3). We have to show that the following laws hold:

1) $A\hat{\oplus}A \approx A, A \otimes A \approx A$; (idempotency) laws
2) $A\hat{\oplus}B \approx B \oplus A, A \otimes B \approx B \otimes A$; (commutativity)
3) $A\hat{\oplus}(B\hat{\oplus}C) \approx (A\hat{\oplus}B)\hat{\oplus}A, A \otimes (B \otimes C) \approx (A \otimes B) \otimes C$; (associativity)
4) $A\hat{\oplus} \perp^0 \approx A, A \otimes \Upsilon \approx A$; (bottom-top laws, see Proposition 36)
5) $A \approx A\hat{\oplus}(A \otimes B), A \approx A \otimes (A\hat{\oplus}B)$; (absorption).

The commutative and idempotency laws hold directly from the definitions of $\hat{\oplus}$ (with Lemma 9) and $\otimes$, and from the results obtained previously. The associative law for '$\hat{\oplus}$' holds from the transitive property of the PO relation '$\preceq$' used in definitions for the sets of simple objects S_A, S_B in Definition 53. The isomorphism (a) $A \otimes (B \otimes C) \approx (A \otimes B) \otimes C$ comes from definition of matching in Theorems 18, and the fact that disjoint union is commutative up to an isomorphism.

Hence, from the fact that isomorphic objects are behaviorally equivalent, we obtain the associativity $A \otimes (B \otimes C) \approx (A \otimes B) \otimes C$ for the matching operator.

Let us show absorption: We have $(A \otimes B) \preceq A$, and hence, from point 4 of Lemma 9, $A\hat{\oplus}(A \otimes B) \approx A$.

Analogously, from point 3 of Lemma 9, $A \preceq (A\hat{\oplus}B)$ and from above, $A \otimes (A\hat{\oplus}B) \approx A$.

For any subset $K \subseteq \mathrm{Ob}_{\mathrm{DB}}$, $\perp^{0} \preceq \inf(K) = \otimes_{A_i \in K} A_i \in \mathrm{Ob}_{\mathrm{DB}}$ and $\Upsilon \succeq \sup(K) = \widehat{\oplus}_{A_i \in K} A_i \in \mathrm{Ob}_{\mathrm{DB}}$ (it holds also for the limit case when $K = \mathrm{Ob}_{\mathrm{DB}}$, with $A_i \subseteq \Upsilon$, i. e., $A_i \preceq \Upsilon$, so that $\sup(\mathrm{Ob}_{\mathrm{DB}}) = \oplus_{A_i \in \mathrm{Ob}_{\mathrm{DB}}} A_i = \Upsilon \in \mathrm{Ob}_{\mathrm{DB}}$).

Consequently, the lattice $(\mathrm{Ob}_{\mathrm{DB}}, \preceq)$ for every subset K has the greatest lower-bound and the least upper-bound, and hence, it is a complete lattice. □

An element B in a lattice $\mathcal{C}$, $B \in \mathcal{C}$, is *compact* iff, for every $X \subseteq \mathcal{C}$ such that $\sup(X)$ exists and $B \preceq \sup(X)$, then there exists a *finite* $X' \subseteq_{\omega} X$ such that $B \preceq \sup(X')$: the set of compact elements in $\mathcal{C}$ is denoted by $\mathrm{Comp}\,\mathcal{C}$. For example, the lattice of subsets of a set is an algebraic lattice where the compact elements are finite sets.

A lattice is *algebraic* if it is *complete* and *compactly generated*: a lattice $(\mathcal{C}, \preceq)$ is compactly generated if every element of $\mathcal{C}$ is a sup of compact elements less then or equal to it, i. e., for every $A \in \mathcal{C}$, $A \approx \sup\{B \in \mathrm{Comp}\,\mathcal{C} \mid B \preceq A\}$. The closed elements, obtained by a closure operator $\mathcal{J}$ from finite subsets, are compact elements. Clearly, the finite lattices are algebraic.

But in our case, the complete database lattice L_{DB} in Proposition 39 is infinite, thus we need to investigate if it is algebraic or not. For example, the subset $[0,1]$ of the real numbers is a complete lattice, but is not algebraic.

The *finite objects* in **DB** are the databases with a finite number of n-ary relations (n is a finite number $n \in \omega$ and the nullary relation $\perp$ is an element of each object in **DB** category); the *extension* of relations is not necessarily finite—in such a case, for a finite object A in **DB**, the object TA is composed of infinite number of relations, i. e., TA is an infinite object.

We will demonstrate that the database lattice is an *algebraic lattice*. One way of producing, and recognizing, complete algebraic lattices is through *closure* operators. We can use this approach because the power-view operator $T : \mathrm{Ob}_{\mathrm{DB}} \to \mathrm{Ob}_{\mathrm{DB}}$ is a closure operator (by Theorem 17 in Section 7.4).

However, the complete lattice derived from a given closure operator must have the meet operator '$\bigwedge$' equal to the set-intersection operator, and, in our case of a complete database lattice $L_{\mathrm{DB}} = (\mathrm{Ob}_{\mathrm{DB}}, \preceq, \otimes, \widehat{\oplus})$, the meet operator is the matching operator: only for the *simple objects* in **DB** it is equal to set-intersection '$\bigcap$'. Consequently, we are able to use the closure operator T in order to demonstrate that the sublattice of L_{DB} composed of only simple database is algebraic.

It is easy to show that the principal properties of the objects and morphisms in the **DB** category are derivable using the properties of the simple objects and simple morphisms between them. For example, a complex object $A \biguplus B$ represents the separation composition of the simple objects (databases) A and B, and hence, the algebraic properties of the complex objects are derivable from the algebraic properties of the simple objects that compose them. The same holds for complex arrows.

The complex arrow $h = [f, g] : A \biguplus B \to C$, $h = f \biguplus g : A \biguplus B \to C \biguplus D$, $h = \langle f, g \rangle : A \to C \biguplus D$ and $h = \lceil f, g \rfloor : A \to C$ are representable in **DB** by the set of their simple ptp arrows f and g.

Consequently, in order to investigate the algebraic database lattice, we can begin from the full subcategory $\underline{\mathbf{DB}}$ of **DB** composed of *only simple* objects, introduced by Definition 52 in Section 7.4 with the object $\underline{Y}$ equal to the union of all simple objects (databases).

By definition, a closed-set system is *algebraic* if $\mathcal{C}$ is closed under unions of upward directed subsets, i. e., for every $S \subseteq \mathcal{C}$, $\bigvee S \in \mathcal{C}$. Equivalently, the closure operator $\mathcal{J}$ on a set Y is algebraic if it satisfies the following "finitary" property: for any subset $X \subseteq Y$,

$$\mathcal{J}(X) = \bigcup\{\mathcal{J}(X') \mid X' \subseteq_{\omega} X\},$$

where $X' \subseteq_{\omega} X$ means that X' is a *finite* subset of X.

Proposition 40. *Let $\mathcal{C} = \mathrm{Ob}_{\underline{DB}_{\mathrm{sk}}} \subset \mathrm{Ob}_{\underline{DB}}$ be the set of all closed objects (w. r. t. the power-view closure operator T) of the $\underline{\mathbf{DB}}$ category. The following properties for a database closure are valid:*

- *A closed-set system $(\underline{Y}, \mathcal{C})$ consists of the "total" closed object (i. e., the top database instance) $\underline{Y} \in \mathcal{C}$ and the set $\mathcal{C}$, which is closed under intersections of arbitrary subsets. That is, for any $K \subseteq \mathcal{C}$, $\bigcap K \in \mathcal{C}$.*
- *The closure operator T is algebraic on the set $\underline{Y}$.*
- *The complete lattice $(\mathcal{C}, \subseteq)$ obtained by this closure operator T is an algebraic lattice with meet $\otimes$ and join $\oplus$ operators. The compact elements of $(\mathcal{C}, \subseteq)$ are closed objects of $\underline{\mathbf{DB}}$ category generated by* finite *simple objects $A \subseteq_{\omega} \underline{Y}$.*
- *The lattice $(\mathrm{Ob}_{\underline{DB}}, \preceq, \otimes, \oplus)$ is an algebraic lattice such that the algebraic closure operator T is a surjective homomorphism from this lattice into the "skeletal" algebraic lattice of closed databases $(\mathcal{C}, \subseteq)$.*

Proof. From Theorem 17 in Section 7.4, $T : \mathrm{Ob}_{\mathrm{DB}} \longrightarrow \mathrm{Ob}_{\mathrm{DB}}$ is a closure operator over the subset $\mathrm{Ob}_{\underline{\mathbf{DB}}}$, and hence, the set of closed (and simple) objects $\mathcal{C}$ is closed under intersection. From Definition 52 in Section 7.4, the total object $\underline{Y}$ is the top object such that for any $A \in \mathcal{C}$ (thus, $A = TA$), $A \subseteq \underline{Y}$. In fact, each object $A \in \mathrm{Ob}_{\underline{\mathbf{DB}}}$ is a subset of $\underline{Y}$ and vice versa, each subset of $\underline{Y}$ (that contains $\perp$) is a database instance, and hence, an object in $\underline{\mathbf{DB}}$ category.

From the Universal algebra theory, it holds that each closure operator and its equivalent closure-set system $(\underline{Y}, \mathcal{C})$ generate a complete lattice $(\mathcal{C}, \subseteq)$, such that for any subset $K \subseteq \mathcal{C}$ of *simple* closed objects $K = \{A_i = TA_i \mid i \in I, A_i$ is closed set of $\underline{\mathbf{DB}}\}$ we obtain:

- The greatest lower bound $\bigwedge K = \bigwedge_{i \in I} A_i = \bigcap_{i \in I} A_i = \bigcap K = \otimes_{i \in I} A_i$ (from Theorem 18 for the simple databases and from $A_i = TA_i$), i. e., meet lattice operator $\bigwedge$ corresponds to the matching operation $\otimes$.
- The least east upper bound $\bigvee K = \bigvee_{i \in I} A_i = T(\bigcup_{i \in I} A_i) = T(\bigcup_{i \in I} TA_i) = \oplus_{i \in I} A_i$ (from Theorem 19 for the simple databases), i. e., the join lattice operator $\bigvee$ corresponds to the merging operation $\oplus$, and hence, for $K = \mathcal{C}$ we obtain $\bigvee \mathcal{C} = T(\bigcup_{A_i \in \mathrm{Ob}_{\underline{\mathbf{DB}}}} A_i) =$ (by Definition 52) $= T\underline{Y} = \underline{Y}$.

Let us demonstrate that T is algebraic, i. e., $T(A) = \bigcup\{T(A') \mid A' \subseteq_\omega A\}$ also for the infinite simple objects $A \subseteq \underline{Y}$. Let $U : \underline{\mathbf{DB}} \longrightarrow \mathbf{K}$ be the unique universal functor (which is a restriction of the functor U to the subcategory $\underline{\mathbf{DB}}$ of only simple objects, described previously in Section 7.4.3). Then, for this infinite simple database A, $U(A) = \{[t_i] \mid t_i \in \mathcal{T}_P A\}$ is an infinite set of the equivalence classes $[t_i]$ of SPJRU algebra terms with an infinite carrier set A of relational tables. From the fact that for all Σ_R-algebra operators $o_i \in \Sigma_R$ with finite $n = \mathrm{ar}(o_i) \leq 2$ and $R_1, \ldots, R_n \in A$, $t_i = \widehat{o_i}(R_1, \ldots, R_n) \in \mathcal{T}_P A$, we can define a finite database $A' = \{R_1, \ldots, R_n\} \subseteq_\omega A$ such that $t_i = \widehat{o_i}(R_1, \ldots, R_n) \in \mathcal{T}_P A'$, $U(A) = \{[t_i] \mid t_i \in \mathcal{T}_P A\} = \bigcup\{[t_i] \mid t_i \in \mathcal{T}_P A', A' \subseteq_\omega A\}$.

Thus, $TA = \{R = \|[t_i]\|_\# \mid t_i \in \mathcal{T}_P A\} = \bigcup\{R = \|[t_i]\|_\# \mid t_i \in \mathcal{T}_P A', A' \subseteq_\omega A\} = \bigcup\{\{R = \|[t_i]\|_\# \mid t_i \in \mathcal{T}_P A'\} \mid A' \subseteq_\omega A\} = \bigcup\{TA' \mid A' \subseteq_\omega A\}$. Hence, T is an algebraic closure operator and, consequently, the lattice $(\mathcal{C}, \subseteq)$ obtained from this closure operator and the closed-set system $(\underline{Y}, \mathcal{C})$ is algebraic. It is well known that the compact elements of such a lattice $(\mathcal{C}, \subseteq)$, obtained from the algebraic closure operator T, are precisely the closed sets $T(A)$, where A is a finite subset of $\underline{Y}$.

The ordering in the lattice $(\mathrm{Ob}_{\underline{DB}}, \preceq, \otimes, \oplus)$ is defined for any two simple databases $A, B \in \mathrm{Ob}_{\underline{DB}}$ by $A \preceq B$ iff $TA \subseteq TB$. Thus, $A \approx B$ iff $TA = TB$, so that $A \approx TA$ with $A \otimes B = TA \bigcap TB$ and $A \oplus B = T(A \bigcup B) = T(TA \bigcup TB)$. From the fact that each nonclosed element A in this lattice is equivalent to the closed element obtained from it ($A \approx TA$), this lattice has the same structure of the algebraic lattice of closed databases $(\mathcal{C}, \subseteq)$. Let us show now that it is also algebraic: in fact, for each $A \in \mathrm{Ob}_{\underline{DB}}$ we have

$$
\begin{aligned}
A \approx TA &= \text{(from the algebraic property of } T\text{)} \\
&= \bigcup\{TB \mid B \subseteq_\omega TA\} \approx T\Big(\bigcup\{TB \mid B \subseteq_\omega TA\}\Big) \\
&\quad (B \text{ is finite, and hence, } B' = TB \in \mathrm{CompC} \text{ is compact element}) \\
&= T\Big(\bigcup\{B' \in \mathrm{CompC} \mid B' \preceq A\}\Big) \\
&= \sup\{B' \in \mathrm{CompC} \mid B' \preceq A\}
\end{aligned}
$$

so that this lattice is compactly generated, and hence, algebraic as well. Consequently, we obtain the surjective homomorphism $T : (\mathrm{Ob}_{\underline{DB}}, \preceq, \otimes, \oplus) \to (\mathcal{C}, \subseteq, \bigcap, T\bigcup)$. □

Now we can extend the lattice $(\mathcal{C}, \subseteq)$ of only closed objects of $\underline{\mathbf{DB}}$ into a lattice of all objects of **DB** category.

Corollary 25. *PO subcategory* $\mathbf{DB}_I \subseteq \mathbf{DB}$, *with* $\mathbf{DB}_I = L_{\mathrm{DB}} = (\mathrm{Ob}_{\mathrm{DB}}, \preceq, \otimes, \widehat{\oplus})$, *is an algebraic lattice.*

Proof. In Proposition 40, it was shown that T is an algebraic closure operator for all *simple* objects in **DB**. Let us show that this closure operator (for all objects in **DB**) is algebraic for the complex objects $A = \biguplus_{1 \leq j \leq m} A_j$, $j \geq 2$ as well. In fact,

$$
\begin{aligned}
T(A) &= T\Big(\biguplus_{1\le j\le m} A_j\Big)\\
&= \biguplus_{1\le j\le m} T(A_j) \quad \text{(from the fact that each } A_j \text{ is a simple object)}\\
&= \biguplus_{1\le j\le m} \Big(\bigcup\{T(A'_j) \mid A'_j \subseteq_\omega A_j\}\Big)\\
&= \bigcup\{(j, T(A'_j)) \mid A'_j \subseteq_\omega A_j, 1 \le j \le m\}\\
&= \bigcup\Big(\biguplus_{1\le j\le m} \{T(A'_j) \mid A'_j \subseteq_\omega A_j\}\Big)\\
&= \bigcup\Big\{\biguplus_{1\le j\le m} T(A'_j) \mid \biguplus_{1\le j\le m} A'_j \subseteq_\omega \biguplus_{1\le j\le m} A_j\Big\}\\
&= \bigcup\{T(A') \mid A' \subseteq_\omega A\}.
\end{aligned}
$$

Thus, the closure operator T is algebraic for all objects in **DB** and, consequently, the complete lattice $L_{\mathrm{DB}} = (\mathrm{Ob}_{\mathrm{DB}}, \preceq, \otimes, \widehat{\oplus})$ is algebraic. Thus, the PO subcategory $\mathbf{DB}_I = L_{\mathrm{DB}}$ (from Theorem 17 in Section 7.4) is algebraic as well. □

The algebraic property is very useful in order to demonstrate the properties of **DB** category: in order to demonstrate the theorems in general, we need to extend the inductive process of a proof beyond ω steps to the transfinite.

Zorn's lemma (equivalent to the axiom of choice of set theory) allows us to do this. The database lattice $L_{\mathrm{DB}} = (\mathrm{Ob}_{\mathrm{DB}}, \preceq, \otimes, \widehat{\oplus})$ is a (nonempty) poset with the property that every chain $K \subseteq \mathrm{Ob}_{\mathrm{DB}}$ (i. e., linearly ordered subset) has an upper bound $\bigvee K = \bigcup K$ (because this poset is *algebraic*) in $\mathrm{Ob}_{\mathrm{DB}}$. Hence, we can apply Zorn's lemma, which asserts that L_{DB} has a maximal element.

Remark. From the fact that $\mathbf{DB}_I = L_{\mathrm{DB}} = (\mathrm{Ob}_{\mathrm{DB}}, \preceq, \otimes, \widehat{\oplus})$ is an algebraic lattice, we obtain for the total object $\Upsilon = T\Upsilon = \bigcup\{TA \mid A \subseteq_\omega \Upsilon\}$, i. e., it is the union of all closed objects generated by only *finite* objects of **DB**, and hence, the union of all *compact* elements of $(\mathcal{C}, \subseteq)$. □

Let ω be the category of natural numbers with the arrows $\le: j \longrightarrow k$ corresponding to the total order relation $j \le k$, i. e., $\omega = \{0 \to 1 \to 2 \to \cdots.\}$. An endofunctor $H : C \longrightarrow D$ is *ω-cocontinuous* if it preserves the colimits of functors $J : \omega \longrightarrow C$, i. e., when $H\mathrm{ColimJ} \simeq \mathrm{ColimHJ}$ (the categories C and D are thus supposed to have these colimits).

Notice that a functor $J : \omega \longrightarrow C$ is a diagram in C of the form $\{C_0 \to C_1 \to C_2 \to \cdots.\}$. For *$\omega$-cocontinuous* endofunctors, the construction of the *initial algebra* is inductive [103].

Proposition 41. *For each simple object A in the category* **DB***, the "merging with A" endofunctor $\sum_A = A \oplus _ : \mathbf{DB} \longrightarrow \mathbf{DB}$ is* ω-cocontinuous.

Proof. Let us consider any chain in **DB** (all arrows are monomorphisms (i. e., "$\preceq$") in a corresponding chain of the $\langle \mathrm{Ob}_{\mathrm{DB}}, \preceq \rangle$ algebraic lattice), that is a diagram $\mathcal{D}$,

$$\perp^0 \preceq_0 \left(\sum_A \perp^0 \right) \preceq_1 \left(\sum_A^2 \perp^0 \right) \preceq_2 \cdots \sum_A^\omega,$$

where $\perp^0 = \{\perp\}$ (with the empty relation $\perp \in A$, so that $\sum_A \perp^0 = A \oplus \perp^0 = T(A \bigcup \perp^0) = TA$) is the initial object in **DB**, with a unique monic arrow $\perp^1 = \preceq_0 : \perp^0 \hookrightarrow (\sum_A \perp^0)$ with $\widetilde{\perp^1} = \perp^0$ and the consecutive arrows $\preceq_n = \sum_A^n \perp^1 : (\sum_A^n \perp^0) \hookrightarrow (\sum_A^{n+1} \perp^0)$ with $\widetilde{\sum_A^n \perp^1} = TA$, for all $n \geq 1$, as representation of a functor $J : \omega \longrightarrow \mathbf{DB}$. The endofunctor $\sum_A$ preserves colimits because it is monotone and $\sum_A^\omega = TA$ is its fixed point, i. e., $\sum_A^\omega = TA = T(A \bigcup TA) = T(A \bigcup \sum_A^\omega) = \sum_A (\sum_A^\omega)$. Thus, the colimit ColimJ $= \sum_A^\omega$ of the base diagram $\mathcal{D}$, given by the functor $J : \omega \longrightarrow \mathbf{DB}$, is equal to ColimJ $= (A \oplus _)^\omega \perp^0 = TA$. Thus, $\sum_A$ ColimJ $= T(A \bigcup \mathrm{ColimJ}) = T(A \bigcup TA) = T(TA) = TA = \mathrm{Colim} \sum_A J$ (where Colim $\sum_A J$ is a colimit of the diagram $\sum_A \mathcal{D}$).

The *ω-cocompleteness* amounts to chain-completeness, i. e., to the existence of least upper bound of *ω-chains*. Consequently, $\sum_A$ is *ω-cocontinuous* endofunctor, i. e., a monotone function, which preserves l. u. b. s of *ω-chains*. □

In what follows, we will pass from lattice based concepts, as l. u. bs of directed subsets, compact subsets and algebraic lattices, to the categorially generalized concepts as directed colimits, finitely presentable (fp) objects, and locally finitely presentable (lfp) categories, respectively:

– A *directed colimit* in **DB** is a colimit of the functor $F : (J, \preceq) \longrightarrow \mathbf{DB}$, where $(J, \preceq)$ is a directed partially ordered set, such that for any two objects $j, k \in J$ there is an object $l \in J$ such that $j \preceq l$, $k \preceq l$, considered as a category. For example, when $J = \mathrm{Ob}_{\mathrm{DB}}$ we obtain the algebraic (complete and compact) lattice, which is an directed PO-set, such that for any two objects $A, B \in J$ there is an object $C \in J$ with $A \preceq C$ and $B \preceq C$ (when $C = \sup(A, B) \in J$).
– An object A is said to be *finitely presentable* (fp), or finitary, if the functor $\mathbf{DB}(A, _) : \mathbf{DB} \to \mathbf{Set}$ preserves directed colimits (or, equivalently, if it preserves filtered colimits). We write $\mathbf{DB}_{fp}$ for the full subcategory of **DB** on the finitely presentable objects: it is essentially small. Intuitively, fp objects are "finite objects," and a category is lfp if it can be generated from its finite objects: a *strong generator* $\mathbf{M}$ of a category is its small full subcategory such that $f : A \longrightarrow B$ is an isomorphism *iff* for all objects C of this subcategory, given a hom-functor $\mathbf{M}(C, _) : \mathbf{M} \longrightarrow \mathbf{Set}$, the following isomorphism of hom-setts $\mathbf{M}(C, f) : \mathbf{M}(C, A) \longrightarrow \mathbf{M}(C, B)$ in **Set** is valid.

From Theorem 1.11 [104], a category is *locally finitely presentable* (lfp) iff it is cocomplete and has a strong generator.

Corollary 26. **DB** *and* $\mathbf{DB}_{\mathrm{sk}}$ *are concrete, locally small and locally finitely presentable categories (lfp).*

Proof. Given any two objects A and B in **DB**, the hom-set $\mathbf{DB}(A,B)$ of all arrows $f : A \longrightarrow B$ corresponds to the directed subset (see Proposition 23) $K = \{\widetilde{f} \mid \perp^0 \preceq \widetilde{f} \preceq A \otimes B\} \subseteq \mathrm{Ob}_{\mathbf{DB}_{\mathrm{sk}}}$, which is bounded algebraic (complete and compact) sublattice of $\mathcal{C}$. Thus, the set of all arrows $f : \Upsilon \longrightarrow \Upsilon$ corresponds to the directed set $K = \{\widetilde{f} \mid \perp^0 \preceq \widetilde{f} \preceq \Upsilon \otimes \Upsilon = \Upsilon\}$, which is equal to the lattice $(\mathrm{Ob}_{\mathbf{DB}_{\mathrm{sk}}}, \preceq)$ (which is a sublattice of the algebraic lattice L_{DB} in Proposition 39, composed of only closed objects in **DB**). Thus, **DB** is *locally small* (has small hom-sets), and, by $\mathbf{DB} \supseteq \mathbf{DB}_{\mathrm{sk}}$, also $\mathbf{DB}_{\mathrm{sk}}$ is locally small.

Let us show that the full subcategory $\mathbf{DB}_{\mathrm{fin}}$, composed of closed objects obtained from *finite* database objects, is a strong generator of **DB**: in fact, if $A \simeq B$, where $A = \biguplus_{1 \leq j \leq m} A_j$, $m \geq 1$ and $B = \biguplus_{1 \leq i \leq k} B_i$, $k \geq 1$ are two *finite* databases (so that from Lemma 4 and Corollary 19 in Section 7.1.1, $k = m$ with a bijection $\sigma : \{1, \ldots, m\} \to \{1, \ldots, k\}$ of simple databases $A_j \simeq B_{\sigma(j)}, 1 \leq j \leq m$) then for all $C = \biguplus_{1 \leq l \leq n} C_l$, $m \geq 1$ in $\mathbf{DB}_{\mathrm{fin}}$, $|\mathbf{DB}_{\mathrm{fin}}(C,A)|$ is the rank of the complete sublattice of $\langle \mathrm{Ob}_{\mathbf{DB}_{\mathrm{sk}}}, \preceq \rangle$ bounded by $\perp^0 \preceq D \preceq C \otimes A$, while $|\mathbf{DB}_{\mathrm{fin}}(C,B)|$ is the rank of the complete sublattice of $\langle \mathrm{Ob}_{\mathbf{DB}_{\mathrm{sk}}}, \preceq \rangle$ bounded by $\perp^0 \preceq D_1 \preceq C \otimes B$. From $A \simeq B$, we deduce $C \otimes A \simeq C \otimes B$, thus $|\mathbf{DB}_{\mathrm{fin}}(C,A)| = |\mathbf{DB}_{\mathrm{fin}}(C,B)|$, i. e., there is a bijection $\upsilon : \mathbf{DB}_{\mathrm{fin}}(C,A) \to \mathbf{DB}_{\mathrm{fin}}(C,B)$, which is an isomorphism in **Set**. Thus, **DB**, which is cocomplete and has this strong generator $\mathbf{DB}_{\mathrm{fin}}$, is an lfp. □

We define a representable functor $\mathbf{DB}(A, _) : \mathbf{DB} \longrightarrow \mathbf{Set}$, such that $\mathbf{DB}(A,B)$ is the hom-set of all morphisms in **DB** from A into B, and for any arrow $g : B \longrightarrow C$, $\mathbf{DB}(A,g)$ is the function such that for any morphism $f \in \mathbf{DB}(A,B)$ we obtain the morphism $h = \mathbf{DB}(A,g)(f) \triangleq g \circ f \in \mathbf{DB}(A,C)$.

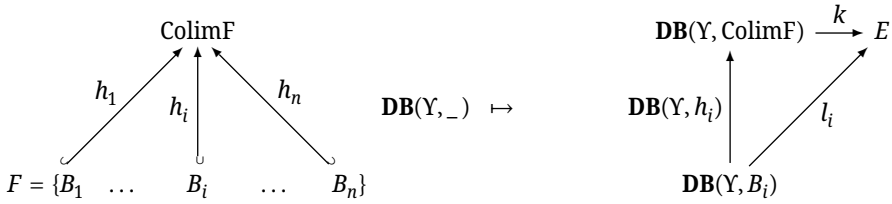
We say that a functor $H : \mathbf{DB} \longrightarrow \mathbf{Set}$ preserves colimits if the image $H\nu : HF \longrightarrow H\mathrm{ColimF}$ for the colimit (ν, ColimF) of a functor $F \in \mathbf{DB}^{\mathbf{J}}$ is a colimiting cone (or cocone) for HF. In this case, we are interested in $H = \mathbf{DB}(\Upsilon, _)$.

Let us show, e. g., that the object Υ is a finitely presentable (fp) (it was demonstrated previously by remark that $\Upsilon = T\Upsilon = \bigcup\{TA \mid A \subseteq_\omega \Upsilon\}$), i. e., the fact that its hom-functor $\mathbf{DB}(\Upsilon, _) : \mathbf{DB} \longrightarrow \mathbf{Set}$ preserves directed colimits.

Proposition 42. *The total object, i. e., the matching monoidal unit* Υ *is a finitely presentable (fp).*

Proof. Let us consider the object ColimF in **DB** (a colimit of the functor $F \in \mathbf{DB}^{\mathbf{J}}$, where F can be seen as a base diagram for this *directed* colimit, composed of a finite number of (possibly complex) objects $B_1, \ldots, B_n$ with PO-arrows "$\preceq$" between them), such that the arrows $h_i : B_i \hookrightarrow \mathrm{ColimF}$ are the monomorphisms corresponding to partial ordering $B_i \preceq \mathrm{ColimF}$ for directed colimits, as specified by Theorem 17 in Section 7.4. These monomorphisms are the components of the cocone (ν, ColimF) where $\nu : F \xrightarrow{\cdot} \triangle \mathrm{ColimF}$ is a natural transformation (and $\triangle$ is a diagonal constant functor) such that its components are $\nu(B_i) = \nu_{B_i} = h_i$, $1 \leq i \leq n$. The translation of this directed colimit diagram from **DB** into **Set** by using the functor $\mathbf{DB}(\Upsilon, _)$, for any cocone monomorphism h_i,

can be represented by the following diagram:



in **DB** in **Set**

Thus, for this directed colimit and a set of all objects in the diagram (i. e., the functor) F, $S = \{B_i \mid B_i \in F\} \subseteq \mathrm{Ob}_{\mathrm{DB}}$, $\mathrm{ColimF} = \sup(S) = \sup\{B_i \mid B_i \in F\} = \sum_{B_i \in F} B_i = \oplus_{B_i \in F} B_i$. Then for each monomorphism $h_i : B_i \preceq \mathrm{ColimF}$, from Proposition 23, any of its simple ptp arrows $h_{ij} \in \frac{h_i}{\boxed{+}}$ is monic, and hence, with $\widetilde{h_{ij}} = TB_i$.

In fact, for any other cocone diagram (ν', D) with a vertex D and with natural transformation $\nu' : F \xrightarrow{\cdot} \triangle D$ that provides the components (ptp arrows) $h'_i = \nu'(B_i) = \nu'_{B_i}$, there is a unique arrow $k : \mathrm{ColimF} \to D$ such that $h'_i = k \circ h_i$, $1 \leq i \leq n$.

This colimit diagram is then translated by the functor $H = \mathbf{DB}(\mathrm{Y}, _)$ in an analogous colimit diagram in **Set** with the cocone-base HF, as presented by the right-hand side of diagram above for any fixed cocone monic arrow $h_i : B_i \hookrightarrow \mathrm{ColimF}$. Let us show that for any other cocone E in **Set**, for the same cocone-base HF (where $H = \mathbf{DB}(\mathrm{Y}, _) : \mathbf{DB} \to \mathbf{Set}$) there is a unique arrow (a function) k from $\mathbf{DB}(\mathrm{Y}, \mathrm{ColimF})$ to the set E (vertex of a cocone E) such that the diagram composed of these two cocones commutes. That is, for each object $\mathbf{DB}(\mathrm{Y}, B_i)$ in the cocone-base HF, we obtain a function l_i, which is a component of cocone E in **Set**:

$$l_i = k \circ \mathbf{DB}(\mathrm{Y}, h_i) : \mathbf{DB}(\mathrm{Y}, B_i) \to E \tag{7.1}$$

for each $1 \leq i \leq n$.

From the functorial property of $H = \mathbf{DB}(\mathrm{Y}, _) : \mathbf{DB} \to \mathbf{Set}$, we have that for the function $\mathbf{DB}(\mathrm{Y}, h_i) : \mathbf{DB}(\mathrm{Y}, B_i) \to \mathbf{DB}(\mathrm{Y}, \mathrm{ColimF})$ it holds that for each DB-morphism $(g : \mathrm{Y} \to B_i) \in \mathbf{DB}(\mathrm{Y}, B_i)$, $\mathbf{DB}(\mathrm{Y}, h_i)(g) = h_i \circ g$ is a DB-morphism with the set of ptp (simple) arrows

$$\frac{h_i \circ g}{\boxed{+}} = \left\{ h_{ij} \circ g_l \mid \mathrm{cod}(g_l) = \mathrm{dom}(h_{ij}), g_l \in \frac{g}{\boxed{+}}, h_{ij} \in \frac{h_i}{\boxed{+}} \right\}$$

where for each monic simple arrow h_{ij}, $\widetilde{h_{ij}} = T(\mathrm{dom}(h_{ij}))$ and

$$\widetilde{g_l} \subseteq T(\mathrm{dom}(g_l)) \bigcap T(\mathrm{cod}(g_l)) \subseteq T(\mathrm{cod}(g_l)) = T(\mathrm{dom}(h_{ij})) = \widetilde{h_{ij}}.$$

That is, $\widetilde{h_{ij} \circ g_l} = \widetilde{h_{ij}} \bigcap \widetilde{g_l} = \widetilde{g_l}$, and hence, by Definition 48 in Section 7.2.3, these two simple morphisms are equivalent arrows in **DB** (up to isomorphism), $h_{ij} \circ g_l \approx g_l$. Conse-

quently, $\frac{h_i \circ g}{\biguplus} \approx \frac{g}{\biguplus}$, i. e., $\mathbf{DB}(\Upsilon, h_i)(g) = h_i \circ g \approx g$, and from (7.1) $l_i(g) = k(\mathbf{DB}(\Upsilon, B_i)(g)) = k(g)$.

Thus, from (7.1), $\forall g \in \mathbf{DB}(\Upsilon, B_i).k(g) = l_i(g)$, for each $1 \leq i \leq n$, i. e., the function k is uniquely defined by the set of cocone functions $l_i, 1 \leq i \leq n$.

Let us consider this problem alternatively by category symmetry, where each DB-morphism f is substituted by its dual object, equal to the information flux $\widetilde{f}$: Each hom-set $\mathbf{DB}(\Upsilon, B_i)$ in **Set** is isomorphic in **Set** to the complete sublattice $(\{\widetilde{f} \mid \widetilde{f} \preceq \Upsilon \otimes B_i\}, \preceq) = (\{\widetilde{f} \mid \widetilde{f} \preceq TB_i\}, \preceq)$ (because each arrow $f : \Upsilon \longrightarrow B_i$ corresponds to the closed object $\widetilde{f} \preceq \Upsilon \otimes B_i = TB_i$). On the other hand, HColimF = $\mathbf{DB}(\Upsilon, \text{ColimF})$ is isomorphic to the complete sublattice $(S, \preceq)$, where $S = \{\widetilde{f} \mid \widetilde{f} \preceq \sum_{B_i \in F} TB_i = \oplus_{B_i \in F} TB_i\}$. Hence, all arrows of the cocone Hv, $\mathbf{DB}(\Upsilon, h_i) : \mathbf{DB}(\Upsilon, B_i) \hookrightarrow \mathbf{DB}(\Upsilon, \text{ColimF})$ are injective functions $(\{\widetilde{f} \mid \widetilde{f} \preceq TB_i\}, \preceq) \subseteq (S, \preceq)$ (also all arrows in the base diagram HF in **Set** are injective functions of the form $(\{\widetilde{f} \mid \widetilde{f} \preceq TB_j\}, \preceq) \subseteq (\{\widetilde{f} \mid \widetilde{f} \preceq TB_k\}, \preceq)$).

Thus, the PO arrows (the monomorphisms) '$\preceq$' in **DB** are translated into injective functions (i. e., the monomorphisms in **Set**), and hence, the directed-colimit diagram in **DB** with ColimF equal to the l. u. b. of all objects in its cocone-base F is translated by the functor $H = \mathbf{DB}(\Upsilon, _)$ into the direct-colimit diagram (with respect to inclusion-PO '$\subseteq$') in **Set** where the $H(\text{ColimF})$ is the l. u. b. with respect to partial ordering $\subseteq$. Consequently, any other E that makes this diagram commutative with the same cocone base HF (composed of all injective functions) from $H(B_i)$ into E for all $1 \leq i \leq n$ must be a bigger set then l. u. b. $H(\text{ColimF})$ and with unique injective function $k : H(\text{ColimF}) \hookrightarrow E$. All arrows of the cocone $E, l_i : (\{\widetilde{f} \mid \widetilde{f} \preceq TB_i\}, \preceq) \longrightarrow E$, must be injective functions (however, with different domains) in order to preserve the commutativity of this colimiting cocone E. Thus, the function $k : (S, \preceq) \longrightarrow E$ is a unique function such that, for any $v \in \langle S, \preceq \rangle$, $k(v) = l_i(v)$ for some $l_i : (\{\widetilde{f} \mid \widetilde{f} \preceq TB_i\}, \preceq) \longrightarrow E$ and $v \in (\{\widetilde{f} \mid \widetilde{f} \preceq TB_i\}, \preceq)$. From HColim = $\mathbf{DB}(\Upsilon, \text{ColimF}) \simeq S$, we can conclude that there is a unique arrow in **Set** from HColimF into E. Consequently, HColim is a colimit in **Set**, i. e., $H = \mathbf{DB}(\Upsilon, _)$ preserves directed colimits, and hence, Υ is finitely presentable. □

Remark. We emphasize the fact that Υ is an fp object for more general considerations of the theory of enriched categories, which will be elaborated in Section 7.5 as, i. e., the demonstration that the monad based on the power-view endofunctor $T : \mathbf{DB} \to \mathbf{DB}$ is an enriched monad. The Kelly–Power theory applies in the case of a symmetric monoidal closed category, which is an lfp and closed category: it is equivalent to demanding that the underlying ordinary category is lfp and that the monoidal structure on this ordinary category restricts to one on its fp objects. For details, see [93, 105], and consider particularly that the unit Υ must be finitely presentable. □

A locally finitely presentable category [106] is the category of models for an essentially *algebraic* theory, which allows operations whose domain is an equationally defined subset of some product of the previously defined domains (the canonical example is a composition in a category, which is defined only on composable, not arbitrary pairs of arrows).

In fact, we deduce from the algebraic (complete and compact) lattice $(\mathrm{Ob}_{\mathrm{DB}}, \preceq)$ that for any simple object A, $A \simeq TA = \oplus\{TB \mid B \subseteq_\omega A\} = \oplus S$ (we recall that $\oplus$ is a generalization in **DB** of the union operation $\bigcup$ for sets and $X \oplus Y = TX \oplus TY$), where the set $S = \{B \mid B \subseteq_\omega A\}$ is upward directed, i. e., for any two finite $B_1, B_2 \subseteq_\omega A$ there is $C = B_1 \bigcup B_2 \in S$ such that $B_1 \preceq C$ and $B_2 \preceq C$, with $TC = B_1 \oplus B_2$. That is, any object in **DB** is generated from finite objects and this generated object is just a directed colimit of these fp objects.

An important consequence of this freedom is that we can express *conditional* equations in the logic for databases.

Another important result, from the fact that **DB** is a complete and cocomplete lfp category, is that it can be used as the category of models for essentially algebraic theory [16, 107] as a relational database theory. Thus, it is a category of models for a finite limit sketch, where sketches are called graph-based logic [108, 109], and it is well known that a relational database *scheme* can readily be viewed, with some inessential abstraction involved, as a sketch. By Liar's theorem, a category **DB** is accessible [110, 111] because it is sketchable.

Remark. A standard application of sketches is used to define a theory of a *single* database scheme, so that objects of this graph-based logic theory are single relations of such a database and arrows between them are used to define the common database functional dependencies, inclusion dependencies and other database constraints. Here, in the case of interdatabase schema mappings, we need to use the whole databases as objects in this lfp **DB** category: the price for this higher level of abstraction is that arrows in **DB** are much more complex than in a standard setting, and hence, are not (simple) functions but the *sets of functions*. □

7.5 Enrichment

It is not misleading, at least initially, to think of an enriched category as being a category in which the hom-sets carry some extra structure (partial order $\preceq$ of algebraic sublattice $(\mathrm{Ob}_{\mathrm{DB}}, \preceq)$ in our case) and in which that structure is preserved by composition. The notion of enriched category [105] is more general and allows for the hom-objects ("hom-sets") of the enriched category to be objects of some monoidal category, traditional called V.

Let us demonstrate that **DB** is a *monoidal closed* category.

Definition 54. For any two objects $B = \biguplus_{1 \le i \le k} B_i$, $k \ge 1$ and $C = \biguplus_{1 \le l \le n} C_l$, $n \ge 1$, the set of all arrows $\mathbf{DB}(B, C)$ from B into C can be represented by unique principal arrow $h : B \to C$ (Proposition 26), such that its set of ptp arrows is

$$\frac{h}{\biguplus} \triangleq \left\{ h_{il} = \bigcup \left\{ f_{il} : B_i \to C_l \mid f_{il} \in \frac{f}{\biguplus}, f \in \mathbf{DB}(B, C) \right\} \neq \emptyset \mid 1 \le i \le k, 1 \le l \le n \right\}.$$

The object C^B is equal to the information flux of this principal arrow. Thus, we define the hom-object

$$C^B \triangleq \widetilde{h} \simeq \widetilde{\biguplus \frac{h}{\biguplus}}.$$

Remark. For any two *simple* objects B and C (when $n = k = 1$), the set of all arrows $S = \{f_1, f_2, \ldots\} : B \to C$, from B into C, can be represented by a unique complex arrow. $\lceil S \rfloor : B \to C$. Thus, from Definition 43 and Lemma 1 in Section 7.1, we obtain the hom-object $C^B \triangleq \widetilde{\lceil S \rfloor} = \oplus\{\widetilde{f_j} \mid f_j \in \mathbf{DB}(B, C)\}$. From Proposition 24, for each simple arrow $f_j \in \mathbf{DB}(B, C)$, $\widetilde{f_j} \subseteq B \otimes C$. Thus, based on the complete lattice $L_{\mathrm{DB}} = (\mathrm{Ob}_{\mathrm{DB}}, \preceq, \otimes, \widehat{\oplus})$ in Proposition 39, the set $\{\widetilde{f_j} \mid f_j \in \mathbf{DB}(B, C)\}$ is a sublattice with the l. u. b. equal to $B \otimes C$, so that the merging (join operator of this lattice) of all elements in this set is equal to its l. u. b. That is, we obtain that $C^B = \oplus\{\widetilde{f_j} \mid f_j \in \mathbf{DB}(B, C)\} = T(\bigcup\{\widetilde{f_j} \mid f_j \in \mathbf{DB}(B, C)\}) = T(\bigcup\{\widetilde{f_j} \mid \widetilde{f_j} \subseteq B \otimes C = TB \bigcap TC\}) = T(TB \bigcap TC) = T(TB) \bigcap T(TC) = TB \bigcap TC = B \otimes C$. □

Consequently, the hom-set $\mathbf{DB}(B, C)$ in **Set** of all morphisms $B \to C$, can be "internalized" into the hom-object C^B in *DB* category by merging of compact elements $A \preceq B \otimes C$ (where $B \otimes C$ is the "distance" between B and C, following Lawvere's idea, for the definition of metric space for the **DB** category [5]).

The aim to "internalize" the hom-sets is justified by necessity to substitute the base category **Set** by the base database category **DB** in the cases as Yoneda embeddings for categories by the contravariant functor $H : \mathbf{C}^{\mathrm{OP}} \to \mathbf{Set}^{\mathbf{C}}$ introduced in technical preliminaries (Section A.2).

Lemma 10. *For any two objects B and C, $C^B \simeq B^C \simeq B \otimes C$.*

Proof. From Definition 54,

$$\begin{aligned} C^B = \widetilde{\biguplus \frac{h}{\biguplus}} &\simeq \biguplus_{1 \le i \le k, 1 \le l \le n} (\oplus\{\widetilde{f_{il}} \mid f_{il} \in \mathbf{DB}(B_i, C_l)\}) \\ &\quad \text{(see the remark above for simple objects)} \\ &= \biguplus_{1 \le i \le k, 1 \le l \le n} B_i \otimes C_l \\ &= B \otimes C \quad \text{(from Definition 40).} \end{aligned}$$

□

Remark. From duality, for any two objects $A = \biguplus_{1 \le j \le m} A_j$, $m \ge 1$ and $B = \biguplus_{1 \le i \le k} B_i$, $k \ge 1$ there is the bijection $_^{\mathrm{OP}} : \mathbf{DB}(A, B) \to \mathbf{DB}(B, A)$, such that for any morphism $f : A \to B$, $f^{\mathrm{OP}} = _^{\mathrm{OP}}(f) : B \to A$, where

$$\frac{f^{\mathrm{OP}}}{\biguplus} = \left\{ f_{ji}^{\mathrm{OP}} \mid f_{ji} : A_j \to B_i \in \frac{f}{\biguplus} \right\},$$

with equal information fluxes $\widetilde{f_{ji}^{\mathrm{OP}}} = \widetilde{f_{ji}}$ (the flux transmitted from simple objects A_j into B_i can be reflexed from B_i to A_j as well). Consequently, by Lemma 10, $A^B \simeq \biguplus_{1\le j\le m,1\le i\le k}(\oplus\{\widetilde{f_{ji}} \mid f_{ji} \in \mathbf{DB}(A_j,B_i)\}) \simeq \biguplus_{1\le i\le k,1\le j\le m}(\oplus\{\widetilde{f'_{ij}} \mid f'_{ij} = f_{ji}^{\mathrm{OP}} \in \mathbf{DB}(B_i,A_j)\}) \simeq B^A \simeq A\otimes B$, i. e., $B\otimes A \simeq A^B \simeq B^A \simeq A\otimes B$.

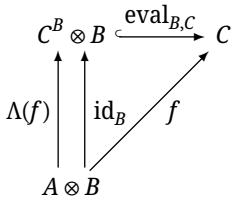
That is, the *cotensor* B^A (i. e., hom-object) of any two objects A and B, which is a particular limit in **DB**, is equal to the corresponding colimit, that is, *tensor product* $A\otimes B$, of these two objects; this fact is based on the duality property of **DB** category.

7.5.1 DB is a V-category enriched over itself

Generally, a monoid M acting on the set $\mathrm{Ob}_{\mathrm{DB}}$ may be seen as general metric space where for any $B \in \mathrm{Ob}_{\mathrm{DB}}$ the distance C^B is a set of $R \in S$ (relations, or views, in our case) whose action sends B to C (this action provides a possibility to pass from the "state" B to "state C of the database "system" of objects in **DB**), as we investigated in the chapter dedicated to operational semantics for the database mapping systems.

A monoidal category is closed if the functor $_\otimes B : \mathbf{DB} \longrightarrow \mathbf{DB}$ has a right adjoint $(_)^B : \mathbf{DB} \longrightarrow \mathbf{DB}$ for every object B, $((_)^B, _\otimes B, \eta_\otimes, \varepsilon_\otimes) : \mathbf{DB} \longrightarrow \mathbf{DB}$, with the counit $\varepsilon_C : C^B\otimes B \longrightarrow C$ called the evaluation at C (denoted by $\mathrm{eval}_{B,C}$).

Theorem 21. *Strictly symmetric idempotent monoidal category* $(\mathbf{DB},\otimes,\Upsilon)$ *is a monoidal biclosed: for every object B, there exists in* **Set** *an isomorphism (a bijection)* $\Lambda : \mathbf{DB}(A\otimes B,C) \cong \mathbf{DB}(A,C^B)$ *such that for any* $f \in \mathbf{DB}(A\otimes B,C)$, $\Lambda(f) \approx f$, *the hom-object* C^B *together with a monomorphism* $\mathrm{eval}_{B,C} : C^B\otimes B \hookrightarrow C$ *the following "exponent" diagram*



commutes, with $f = \mathrm{eval}_{B,C}\circ(\Lambda(f)\otimes \mathrm{id}_B)$.

Proof. Let us consider the simple objects A, B, C and a morphism

$$f = \alpha * \big(\mathrm{MakeOperads}(\mathcal{M})\big) : A\otimes B \to C,$$

obtained from a sketch's mapping $\mathbf{M} : \mathcal{A}\bigcap_{\alpha}\mathcal{B} \to \mathcal{C}$ with $A = \alpha^*(\mathcal{A})$, $B = \alpha^*(\mathcal{B})$ and $C = \alpha^*(\mathcal{C})$ obtained from the schemas $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$. Thus, $\widetilde{f} \subseteq T(A\otimes B)\bigcap TC = TA\bigcap TB\bigcap TC$.

Then we define the schema mapping $\mathcal{M}_1 = \{\Phi\} : \mathcal{A}\bigcap_{\alpha}\mathcal{A} \to \mathcal{B}\bigcap_{\alpha}\mathcal{C}$ with SOtgd Φ equal to conjunction $\bigwedge\{\forall\mathbf{x}(r(\mathbf{x}) \Rightarrow r(\mathbf{x})) \mid r \in \mathcal{A}\bigcap_{\alpha}\mathcal{A}$ and $\alpha(r) \in \widetilde{f} \subseteq \alpha^*(\widehat{\mathcal{A}}) = TA\}$ with the sketch's mapping $\mathbf{M}_1 = \{1_r \in O(r,r) \mid \alpha(r) \in \widetilde{f} \subseteq TA\}$, where $\mathcal{A}\bigcap_{\alpha}\mathcal{B}$ is the α-intersection

schema (defined in Definition 70 in Section A.3.5) with $\alpha^*(\mathcal{A}\bigcap_{\alpha}\mathcal{A}) = TA$, and hence, its flux is $\widetilde{\alpha^*(\mathbf{M}_1)} = \mathrm{Flux}(\alpha, \mathbf{M}_1) = \widetilde{f}$.

Then we define the **DB** morphism

$$\Lambda(f) = \alpha^*(\mathbf{M}_1) \circ \mathrm{is}_A : A \to \alpha\left(\mathcal{B}\bigcap_{\alpha}\mathcal{C}\right) = B \otimes C = C^B.$$

This morphism from A into C^B is uniquely defined by the morphism $f : A \otimes B \to C$: in fact, its flux is uniquely determined by

$$\widetilde{\Lambda(f)} = \widetilde{\alpha^*(\mathbf{M}_1) \circ \mathrm{is}_A} = \widetilde{\alpha^*(\mathbf{M}_1)} \bigcap \widetilde{\mathrm{is}_A} = \widetilde{f} \bigcap TA = \widetilde{f}.$$

Analogously, we define the schema mapping $\mathcal{M}_2 = \{\Psi\} : \mathcal{B}\bigcap_{\alpha}\mathcal{C} \to \mathcal{C}$ with SOtgd Ψ equal to the conjunction $\bigwedge\{\forall \mathbf{x}(r(\mathbf{x}) \Rightarrow r(\mathbf{x})) \mid r \in \mathcal{B}\bigcap_{\alpha}\mathcal{C} \text{ and } \alpha(r) \in \alpha^*(\mathcal{B}\bigcap_{\alpha}\mathcal{C}) = TB \bigcap TC\}$ with the sketch's mapping

$$\mathbf{M}_2 = \mathrm{MakeOperads}\left\{1_r \in O(r,r) \mid \alpha(r) \in TB \bigcap TC\right\}.$$

Hence, we define the **DB** morphism $\mathrm{eval}_{B,C} = \alpha^*(\mathbf{M}_2) : C^B \otimes B = \alpha * (\mathcal{B}\bigcap_{\alpha}\mathcal{C}) \to C$, which is a monomorphism from the fact that $\widetilde{\mathrm{eval}_{B,C}} = TB \bigcap TC = C^B \otimes B$.

Consequently, we obtain for the commutativity of this exponential diagram, i. e., $\widetilde{f} = \widetilde{\mathrm{eval}_{B,C} \circ (\Lambda(f) \otimes \mathrm{id}_B)} = \widetilde{\mathrm{eval}_{B,C}} \bigcap (\widetilde{\Lambda(f)} \otimes \widetilde{\mathrm{id}_B}) = \widetilde{\mathrm{eval}_{B,C}} \bigcap (\widetilde{\Lambda(f)} \otimes \widetilde{\mathrm{id}_B}) = B \otimes C \bigcap \widetilde{\Lambda(f)} \bigcap TB = TB \bigcap TC \bigcap \widetilde{\Lambda(f)} = \widetilde{\Lambda(f)}$ (from the fact that $\widetilde{\Lambda(f)} \subseteq C^B = TB \bigcap TC$). Thus, $f = \mathrm{eval}_{B,C} \circ (\Lambda(f) \times \mathrm{id}_B)$ iff $\Lambda(f) \approx f$.

Λ is a bijection because $\mathbf{DB}(A \otimes B, C) = \{g \mid {\perp^0} \subseteq \widetilde{g} \subseteq A \otimes B \otimes C\} \cong \{\widetilde{g} \mid \widetilde{g} \in K\}$, where K is a bounded algebraic sublattice (of closed objects) of the lattice $(\mathcal{C}, \subseteq)$ in Proposition 40 and $\cong$ denotes a bijection, i. e., $K = \{a \mid a \in \mathcal{C} \text{ and } a \subseteq TA \bigcap TB \bigcap TC\}$. Also, $\mathbf{DB}(A, C^B) = \mathbf{DB}(A, B \otimes C) \cong K$, thus $|\mathbf{DB}(A \otimes B, C)| = |\mathbf{DB}(A, C^B)| = |K|$. Consequently, Λ is a bijection such that for any $f \in \mathbf{DB}(A \otimes B, C)$, $\widetilde{\Lambda(f)} = \widetilde{f} \in K$, i. e., $\Lambda(f) \approx f$.

Let us consider now the complex objects $A = \biguplus_{1 \le j \le m} A_j$, $m \ge 1$, $B = \biguplus_{1 \le i \le k} B_i$, $k \ge 1$ and $C = \biguplus_{1 \le l \le n} C_l$, $n \ge 1$. Then $\mathrm{eval}_{B,C}$ is the monomorphism $\mathrm{in} : C \otimes B \hookrightarrow C$, obtained by Theorem 17 in Section 7.4 from the PO relation (from the isomorphism $C^B \otimes B \simeq C \otimes B \otimes B$), $C^B \otimes B \approx C \otimes B \otimes B$ (from $C \otimes B \preceq B$) $\approx C \otimes B \preceq C$ in the complete lattice $L_{\mathrm{DB}} = (\mathrm{Ob}_{\mathrm{DB}}, \preceq, \otimes, \widehat{\oplus})$ (in Proposition 39), with $\frac{in}{\biguplus} = \{\mathrm{in}_{li} : C_l \otimes B_i \hookrightarrow C_l \mid 1 \le l \le n, 1 \le i \le k\}$.

For a given complex arrow $f : A \otimes B \to C$, we define the mapping Λ such that

$$\frac{f}{\biguplus} \mapsto \left\{f'_{jli} : A_j \to C_l \otimes B_i \mid (f_{jil} : A_j \otimes B_i \to C_l) \in \frac{f}{\biguplus} \quad \text{and} \quad \widetilde{f'_{jli}} = \widetilde{f_{jil}}\right\},$$

and its inverse Λ^{-1} such that

$$\frac{\Lambda(f)}{\biguplus} \mapsto \left\{f_{jil} : A_j \otimes B_i \to C_l \mid (f'_{jli} : A_j \to C_l \otimes B_i) \in \frac{\Lambda(f)}{\biguplus} \quad \text{and} \quad \widetilde{f_{jil}} = \widetilde{f'_{jli}}\right\},$$

and hence Λ is a bijection.

Let us show the commutativity $f = in \circ (\Lambda(f) \otimes \mathrm{id}_B)$ of the diagram, i. e., that $\frac{f}{\biguplus} = \frac{in \circ (\Lambda(f) \otimes \mathrm{id}_B)}{\biguplus}$. In fact, for each $(\mathrm{in}_{li} \circ (f'_{jli} \otimes \mathrm{id}_{B_i}) : A_j \otimes B_i \to C_l) \in \frac{in \circ (\Lambda(f) \otimes \mathrm{id}_B)}{\biguplus}$, $\mathrm{in}_{li} \circ \widetilde{(f'_{jli} \otimes \mathrm{id}_{B_i})} = (TC_l \bigcap TB_i) \bigcap (\widetilde{f'_{jli}} \bigcap TB_i) =$ (from $\widetilde{f'_{jli}} \subseteq TA_j \bigcap TB_i \bigcap TC_l$) $= \widetilde{f'_{jli}} = \widetilde{f_{jil}}$, for the corresponding $(f_{jil} : A_j \otimes B_i \to C_l) \in \frac{f}{\biguplus}$. Consequently, $\mathrm{in}_{li} \circ (f'_{jli} \otimes \mathrm{id}_{B_i}) = f_{jil}$ for each ptp arrow in $in \circ (\Lambda(f) \otimes \mathrm{id}_B)$ and vice versa. Hence, $f = in \circ (\Lambda(f) \otimes \mathrm{id}_B) = \mathrm{eval}_{B,C} \circ (\Lambda(f) \otimes \mathrm{id}_B)$.

Consequently, **DB** is closed and symmetric, i. e., a biclosed category. □

We have seen that all objects in **DB** are finitely representable (from Corollary 26 and Proposition 42). Let us denote the representable functor $\mathbf{DB}(\Upsilon, _)$ by $V = \mathbf{DB}(\Upsilon, _) : \mathbf{DB} \longrightarrow \mathbf{Set}$. By putting $A = \Upsilon$ in $\Lambda(f) : A \to C^B$ of the "exponent" diagram in Theorem 21, and by using the isomorphism $\beta : \Upsilon \otimes B \simeq B$, we get a natural isomorphism in **Set** (a bijection Λ), $\mathbf{DB}(B, C) \simeq V(C^B) = \mathbf{DB}(\Upsilon, C^B)$.

Then C^B is exhibited as a lifting through V of the hom-set $\mathbf{DB}(B, C)$ (i. e., in the case when B and C are the simple objects, the hom-object C^B is a set of all views, which gives a possibility to pass from a "state" B to a "state" C). It is called the *internal hom* of B and C.

By putting $B = \Upsilon$ in $\Lambda(f) : A \to C^B$ of the "exponent" diagram in Theorem 21, and by using the isomorphism $\gamma : A \otimes \Upsilon \simeq A$ we deduce a natural isomorphism $i : C \simeq C^{\Upsilon}$ (it is obvious, from the fact that $C^{\Upsilon} \simeq C \otimes \Upsilon \simeq C$).

The fact that a monoidal structure is closed means that we have an internal Hom functor, $(_)^{(_)} : \mathbf{DB}^{\mathrm{OP}} \times \mathbf{DB} \to \mathbf{DB}$, which "internalizes" the external Hom functor, $\mathrm{Hom} : \mathbf{DB}^{\mathrm{OP}} \times \mathbf{DB} \to \mathbf{Set}$, such that for any two objects A and B, the hom-object $B^A = (_)^{(_)}(A, B)$, represents the hom-set $\mathrm{Hom}(A, B)$ (i. e., the set of all morphisms from A to B).

Monoidal closed categories generalize the Cartesian closed ones in that they also posses exponent objects B^A, which "internalize" the hom-sets. One may then ask if there is a way to "internally" describe the behavior of functors on morphisms. That is, given a monoidal closed category C and a functor $F : C \longrightarrow C$, consider, say, $f \in C(A, B)$ then $F(f) \in C(F(A), F(B))$. Since the hom-object B^A and $F(B)^{F(A)}$ represent hom-sets $C(A, B)$ and $C(F(A), F(B))$ in C, one may study the conditions under which F is "represented" by morphism in $C(B^A, F(B)^{F(A)})$, for each A and B.

Theorem 22. *The endofunctor $T : \mathbf{DB} \longrightarrow \mathbf{DB}$ is closed.*

DB *is a V-category enriched over itself, with the composition law monomorphism $m_{A,B,C} : C^B \otimes B^A \hookrightarrow C^A$ and identity element (epimorphism) $j_A : \Upsilon \twoheadrightarrow A^A$, which "picks up" the identity in A^A.*

The monad (T, η, μ) is an enriched monad, and hence, **DB** *is an object of V-cat and endofunctor $T : \mathbf{DB} \to \mathbf{DB}$ is an arrow of V-cat.*

Proof. Let us show that the arrows introduced by this proposition satisfy the morphisms in the definition of the **DB** category; see Theorem 10 in Section 7.1. For a given database schemas $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ and am R-algebra α such that $A = \alpha^*(\mathcal{A})$, $B = \alpha^*(\mathcal{B})$, $C = \alpha^*(\mathcal{C})$, we

have that there exist the α-intersection schemas $\mathcal{A}\bigcap\limits_{\alpha}\mathcal{C}$ and $(\mathcal{A}\bigcap\limits_{\alpha}\mathcal{C})\bigcap\limits_{\alpha}\mathcal{B}$ so that $\alpha^*(\mathcal{A}\bigcap\limits_{\alpha}\mathcal{C}) = A\otimes C = C^A$ and

$$\alpha^*\left(\left(\mathcal{A}\bigcap_{\alpha}\mathcal{C}\right)\bigcap_{\alpha}\mathcal{B}\right) = TA\bigcap TC\bigcap TB = A\otimes C\otimes B = C^B\otimes B^A.$$

Thus, we may define a schema mapping $\mathcal{M} = \{\Phi\} : (\mathcal{A}\bigcap\limits_{\alpha}\mathcal{C})\bigcap\limits_{\alpha}\mathcal{B} \to \mathcal{A}\bigcap\limits_{\alpha}\mathcal{C}$, where Ψ is a SOtgd conjunction $\bigwedge\{\forall\mathbf{x}(r(\mathbf{x})\Rightarrow r(\mathbf{x}))\mid r\in(\mathcal{A}\bigcap\limits_{\alpha}\mathcal{C})\bigcap\limits_{\alpha}\mathcal{B}\}$, and $\mathbf{M} = \text{MakeOperads}(\mathcal{M}) = \{1_r \mid r\in(\mathcal{A}\bigcap\limits_{\alpha}\mathcal{C})\bigcap\limits_{\alpha}\mathcal{B}\}\bigcup\{1_{r_\emptyset}\}$, so we obtain a monomorphism

$$\begin{aligned} m_{A,B,C} &= \alpha^*(\mathbf{M}) \\ &= \left\{1_R : R\to R \mid R\in TA\bigcap TC\bigcap TB\right\}\bigcup\{q_\perp\} : C^B\otimes B^A \hookrightarrow C^A. \end{aligned}$$

In the case of complex objects A, B and C, the $m_{A,B,C}$ is uniquely determined by the PO relation $C^B\otimes B^A \preceq C^A$ as specified by Theorem 17 in Section 7.4. In fact, from $C^B\otimes B^A \simeq (B\otimes C)\otimes(A\otimes B)$, $C^B\otimes B^A \approx (B\otimes C)\otimes(A\otimes B) \approx (C\otimes B)\otimes(B\otimes A) \approx$ (from $B\otimes B\approx B$) $\approx C\otimes B\otimes A \approx B\otimes(A\otimes C) \preceq$ (from the complete lattice $L_{\text{DB}} = (\text{Ob}_{\text{DB}}, \preceq, \otimes, \widehat{\oplus})$ in Proposition 39) $\preceq A\otimes C$ (from $A\otimes C\simeq C^A$) $\approx C^A$.

The identity element j_A is defined by a schema mapping $\mathcal{M}_1 = \{\Psi\} : \widehat{\Upsilon}\to\mathcal{A}\bigcap\limits_{\alpha}\mathcal{A}$, where Ψ is a SOtgd $\bigwedge\{\forall\mathbf{x}(r(\mathbf{x})\Rightarrow r(\mathbf{x}))\mid r\in\mathcal{A}\bigcap\limits_{\alpha}\mathcal{A}\}$, and hence,

$$j_A = \alpha^*(\text{MakeOperads}(\mathcal{M}_1)) = \{1_R : R\to R\mid R\in TA\}\bigcup\{q_\perp\} : \Upsilon\twoheadrightarrow A^A.$$

In the case when $A = \biguplus_{1\le j\le m} A_j$, $m\ge 2$ is a complex object, with $A^A = A\otimes A$ so that j_A is defined by the set of ptp arrows $\frac{j_A}{\biguplus} = \frac{\text{id}_A\otimes\text{id}_A}{\biguplus}$ with $\widetilde{j_A} \simeq \widetilde{\biguplus\frac{j_A}{\biguplus}} = \biguplus\widetilde{\frac{\text{id}_A\otimes\text{id}_A}{\biguplus}} = A\otimes A$, and consequently, by Proposition 25 in Section 7.1, j_A is an epimorphism.

Analogously, for all other morphisms used in this proof a schema mapping can be provided from which they are derived by an R-algebra α.

It is easy to verify that for each pair of objects (i. e., databases) A and B in **DB** there exists $f_{AB}\in\mathbf{DB}(B^A, (TB)^{TA})$, called "an *action* of T on B^A," such that for all $g\in\mathbf{DB}(A,B)$ the following is valid:

$$f_{AB}\circ\Lambda(g\circ\beta(A)) = \Lambda(T(g)\circ\beta(TA)) : \Upsilon\longrightarrow(TB)^{TA},$$

where $\beta : \Upsilon\otimes_\longrightarrow I_{\text{DB}}$ is a left identity natural transformation of a monoid $(\mathbf{DB}, \otimes, \Upsilon, \alpha, \beta, \gamma)$, thus $\beta(A) = \text{is}_A^{-1} : \Upsilon\otimes A\to A$ (where $\Upsilon\otimes A = TA$ from Theorem 18) and $\beta(TA) = \text{id}_{TA}$. In fact, we take $f_{AB} = \text{id}_{B^A}$ (the action of T on $f : A\to B$ is equal to $Tf : TA\to TB$ with $\widetilde{Tf} = \widetilde{f}$), which is an identity action: moreover, $B^A\simeq A\otimes B = TA\otimes TB\simeq TB^{TA}$. Hence, by considering that for $A = \biguplus_{1\le j\le m}$, $m\ge 1$, $B = \biguplus_{1\le i\le k}$, $k\ge 1$ and $\text{is}_A^{-1} : TA\to A$ with $\frac{\text{is}_A^{-1}}{\biguplus} = \{\text{is}_{A_j}^{-1} : TA_j\to A_j \mid 1\le j\le m\}$, we obtain (see the proof of Theorem 21):

$$\frac{\Lambda(T(g))}{\biguplus} = \left\{ T(g_{ji}) \mid (g_{ji} : A_j \to B_i) \in \frac{g}{\biguplus} \right\}$$

and

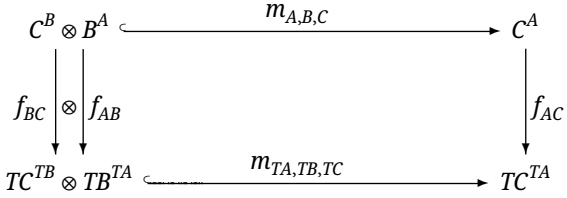
$$\begin{aligned}
\frac{\Lambda(g \circ \mathrm{is}_A^{-1})}{\biguplus} &= \left\{ g'_{ji} \mid (g_{ji} \circ \mathrm{is}_{A_j}^{-1} : TA_j \to B_i) \in \frac{g \circ \mathrm{is}_A^{-1}}{\biguplus}, \widetilde{g'_{ji}} = \widetilde{g_{ji} \circ \mathrm{is}_{A_j}^{-1}} \right\} \\
&= \left\{ g'_{ji} \mid g_{ji} \in \frac{g}{\biguplus}, \widetilde{g'_{ji}} = \widetilde{g_{ji}} \bigcap \widetilde{\mathrm{is}_{A_j}^{-1}} \right\} \\
&= \left\{ g'_{ji} \mid g_{ji} \in \frac{g}{\biguplus}, \widetilde{g'_{ji}} = \widetilde{g_{ji}} \bigcap TA_j \right\} \quad \left(\text{from } \widetilde{g_{ji}} \subseteq A_j \otimes B_i = TA_j \bigcap TB_i \subseteq TA_j \right) \\
&= \left\{ g'_{ji} \mid g_{ji} \in \frac{g}{\biguplus}, \widetilde{g'_{ji}} = \widetilde{g_{ji}} \right\}.
\end{aligned}$$

Thus, for each $g_{ji} \in \frac{g}{\biguplus}$, $\widetilde{g'_{ji}} = \widetilde{g_{ji}} = T(g_{ji})$, i. e., $g'_{ji} = T(g_{ji})$ and, consequently,

$$\Lambda(g \circ \mathrm{is}_A^{-1}) = \Lambda(T(g)). \tag{7.2}$$

Hence, $f_{AB} \circ \Lambda(g \circ \beta(A)) = \mathrm{id}_{B^A} \circ \Lambda(g \circ \mathrm{is}_A^{-1}) = \Lambda(g \circ \mathrm{is}_A^{-1}) =$ (from (7.2)) $= \Lambda(T(g)) = \Lambda(T(g) \circ \mathrm{id}_{TA}) = \Lambda(T(g) \circ \beta(TA))$. Consequently, T is a closed endofunctor.

The composition law $m_{A,B,C}$ may be equivalently represented by a natural transformation $m : (B \otimes _) \otimes (_ \otimes B) \xrightarrow{\cdot} \otimes$ and an identity element j_A by natural transformation $j : Y \xrightarrow{\cdot} \otimes \circ \triangle$, where $\triangle : \mathbf{DB} \longrightarrow \mathbf{DB} \times \mathbf{DB}$ is a diagonal functor, while $Y : \mathbf{DB} \longrightarrow \mathbf{DB}$ is a constant endofunctor, $Y(A) \triangleq \mathrm{Y}$ for any A and $Y(f) \triangleq \mathrm{id}_{\mathrm{Y}}$ for any arrow f in **DB**. It is easy to verify that two coherent diagrams (associativity and unit axioms) commute, and hence, **DB** is enriched over itself V-category (as, i. e., **Set** category). Functor T is a V-functor: for each pair of objects A and B there exists an identity map (see above) $f_{AB} : B^A \longrightarrow (TB)^{TA}$, subject to the compatibility with composition m and with the identities expressed by the commutativity of the diagram



i. e., $f_{AC} \circ m_{A,B,C} = m_{TA,TB,TC} \circ (f_{BC} \otimes f_{AB})$ and $j_{TA} = f_{AB} \circ j_A$. From the fact that f_{BC} and f_{AB} are the identity arrows, $(f_{BC} \otimes f_{AB})$ is identity as well, so it is enough to show that $m_{TA,TB,TC} = m_{A,B,C}$. In fact, for any $A = \biguplus_{1 \le j \le m}$, $m \ge 1$, $B = \biguplus_{1 \le i \le k}$, $k \ge 1$ and $C = \biguplus_{1 \le l \le n}$, $n \ge 1$, and from $C^B \simeq B \otimes C$, $B^A \simeq A \otimes B$, we obtain $\frac{m_{A,B,C}}{\biguplus} = \{\mathrm{in}_{ilji_1} : (B_i \otimes C_l) \otimes (A_j \otimes B_{i_1}) \hookrightarrow A_j \otimes C_l \mid 1 \le j \le m, 1 \le i, i_1 \le k, 1 \le l \le n\} = \{\mathrm{in}_{ilji_1} : (TB_i \bigcap TC_l) \bigcap (TA_j \bigcap TB_{i_1}) \hookrightarrow TA_j \bigcap TC_l \mid 1 \le j \le m, 1 \le i, i_1 \le k, 1 \le l \le n\} \{\mathrm{in}_{ilji_1} : (T(TB_i) \bigcap T(TC_l)) \bigcap (T(TA_j) \bigcap T(TB_{i_1})) \hookrightarrow$

$T(TA_j) \bigcap T(TC_l) \mid 1 \leq j \leq m, 1 \leq i, i_1 \leq k, 1 \leq l \leq n\} = \{\mathrm{in}_{ilji_1} : (TB_i \otimes TC_l) \otimes (TA_j \otimes TB_{i_1}) \hookrightarrow TA_j \otimes TC_l \mid 1 \leq j \leq m, 1 \leq i, i_1 \leq k, 1 \leq l \leq n\} = \frac{m_{TA,TB,TC}}{\downarrow}$, i. e., $m_{TA,TB,TC} = m_{A,B,C}$.

It is easy to verify that also natural transformations $\eta : I_{\mathrm{DB}} \xrightarrow{\cdot} T$ and $\mu : T^2 \xrightarrow{\cdot} T$ satisfy the V-naturality condition (V-natural transformation η and μ are an $\mathrm{Ob}_{\mathrm{DB}}$-indexed family of components $\delta_A : \Upsilon \nrightarrow TA$ in **DB**; for η, $\delta_A : \Upsilon \nrightarrow (TA)^A$, $(TA)^A \approx TA$ while for μ, $\delta_A : \Upsilon \nrightarrow TA^{T^2A}$, $(TA)^{T^2A} \approx TA$). This map f_{AB} is equal also for the endofunctor identity I_{DB}, and for the endofunctor T^2, because $B^A \simeq A \otimes B \simeq (TB)^{TA} = (T^2B)^{T^2A}$. □

In fact, each monoidal closed category is itself a V-category: hom-sets from A to B are defined as "internalized" hom-objects (cotensors) B^A. The composition is given by the image of the bijection $\Lambda : \mathbf{DB}(D \otimes A, C) \simeq \mathbf{DB}(D, C^A)$, where $D = C^B \otimes B^A$, of the arrow $\varepsilon_B \circ (\mathrm{id}_{C^B} \otimes \varepsilon_A) \circ \alpha_{C^B,B^A,A}$, and hence, $m_{A,B,C} = \Lambda(\varepsilon_B \circ (\mathrm{id}_{C^B} \otimes \varepsilon_A) \circ \alpha_{C^B,B^A,A}) = \Lambda(\mathrm{eval}_{B,C} \circ (\mathrm{id}_{C^B} \otimes \mathrm{eval}_{A,B}) \circ \alpha_{C^B,B^A,A})$, i. e., a monomorphism. The identities are given by the image of the isomorphism $\beta_A : \Upsilon \otimes A \longrightarrow A$, under the bijection $\Lambda : \mathbf{DB}(\Upsilon \otimes A, A) \simeq \mathbf{DB}(\Upsilon, A^A)$, and hence, $j_A = \Lambda(\beta_A) : \Upsilon \nrightarrow A^A$ (j_A that is an epimorphism).

Moreover, for a V-category **DB**, the following isomorphism (which extends the tensor-cotensor isomorphism Λ of exponential diagram in Theorem 21) is valid in all enriched Lawvere theories [107], $\mathbf{DB}(D \otimes A, C) \simeq \mathbf{DB}(D, C^A) \simeq \mathbf{DB}(A, \mathbf{DB}(D, C))$.

Finally, from the fact that **DB** is an lfp category enriched over the lfp symmetric monoidal closed category with a tensor product $\otimes$ (i. e., the matching operator for databases), and the fact that T is a finitary enriched monad on **DB**, from the Kelly–Power theorem we have that **DB** admits a presentation by operations and equations, which what was implicitly assumed in the definition of this power-view operator (see also [61, 112]).

7.5.2 Internalized Yoneda embedding

As we have seen in Section 7.5.1, the fact that a monoidal structure is closed means that we have an internal hom-functor $(_)^{(_)} : \mathbf{DB}^{\mathrm{OP}} \times \mathbf{DB} \to \mathbf{DB}$, which "internalizes" the external hom-functor, $\mathrm{hom} : \mathbf{DB}^{\mathrm{OP}} \times \mathbf{DB} \to \mathbf{Set}$, such that for any two objects A and B, the hom-object $B^A = (_)^{(_)}(A, B)$ represents the hom-set $\mathrm{hom}(A, B) = \mathbf{DB}(A, B)$ (the set of all morphisms from A to B). We may apply the Yoneda embedding to the **DB** category by the contravariant functor $H : \mathbf{DB}^{\mathrm{OP}} \to \mathbf{Set}^{\mathbf{DB}}$, such that for any object A we obtain a functor $H_A = \mathbf{DB}(A, _) : \mathbf{DB} \to \mathbf{Set}$, and hence, for any object B in **DB**, $H_A(B) = \mathrm{hom}(A, B) = \mathbf{DB}(A, B)$ is a hom-set of all morphisms in **DB** between A and B, and for any morphism $f : B \to C$ in **DB** the function $H_A(f) = \mathbf{DB}(A, f) : \mathbf{DB}(A, B) \to \mathbf{DB}(A, C)$ is a composition $H_A(f) = f \circ _$.

The internalized Yoneda embedding can be obtained by replacing the base **Set** category with the base database category **DB**, and hence, this embedding is provided by the contravariant functor $\overline{H} : \mathbf{DB}^{\mathrm{OP}} \to \mathbf{DB}^{\mathbf{DB}}$. So, we internalize a hom-set $H_A(B) =$

$\hom(A,B) = \mathbf{DB}(A,B)$ in **Set** by the (co)tensor $\overline{H}_A(B) = A \otimes B$ in **DB**, and we internalize a composition function $H_A(f) = f \circ _$ in **Set** by the (co)tensor composition of morphisms in **DB**, $\overline{H}_A(f) = \mathrm{id}_A \otimes f : A \otimes B \to A \otimes C$.

Proposition 43. *For any object A in* **DB** *with a monomorphisms $\mathrm{in}_A : A \hookrightarrow \Upsilon$, the natural transformation $\phi : \overline{H}_A \to T$ is monomorphic and such that $\phi = \mathrm{in}_A \otimes \mathrm{id}_{\{_\}}$.*

Proof. For any object A, from the internalized Yoneda embedding $\overline{H} : \mathbf{DB}^{\mathrm{OP}} \to \mathbf{DB}^{\mathbf{DB}}$, we obtain the covariant functor $\overline{H}_A = \overline{H}(A)$ such that $\overline{H}_A = A \otimes _ : \mathbf{DB} \to \mathbf{DB}$. Consequently, for the covariant power-view functor $T : \mathbf{DB} \to \mathbf{DB}$, the natural transformation $\phi : \overline{H}_A \xrightarrow{\cdot} T$ is well-defined and monomorphic if, for *every* arrow $f : B \to C$ in **DB**, the following diagram commutes:

$$\begin{array}{ccc} \overline{H}_A(B) = A \otimes B & \xrightarrow{\overline{H}_A(f) = \mathrm{id}_A \otimes f} & A \otimes C = \overline{H}_A(C) \\ \downarrow{\scriptstyle \phi_B} & & \downarrow{\scriptstyle \phi_C} \\ \Upsilon \otimes B = TB & \xrightarrow{Tf} & TC = \Upsilon \otimes C \end{array}$$

Notice that, from the fact that for each $A \in \mathrm{OB}_{\mathrm{DB}}$, $A \preceq \Upsilon$ and $\overline{H}_\Upsilon = \Upsilon \otimes _ = T$. Thus, $\phi : \overline{H}_A \to \overline{H}_\Upsilon$, i. e., for any object B, $\phi_B = \mathrm{in}_A \otimes \mathrm{id}_B : A \otimes B \hookrightarrow \Upsilon \otimes B$ and $\phi_C = \mathrm{in}_A \otimes \mathrm{id}_c$, where $\mathrm{in}_A : A \hookrightarrow \Upsilon$ is the unique monomorphism (in this case the inclusion) of A into the total object Υ, so that the diagram above can be reduced to the following commutative diagram in **DB** for any $f \in \mathbf{DB}(B,C)$:

$$\begin{array}{ccc} \overline{H}_A(B) = A \otimes B & \xrightarrow{\overline{H}_A(f) = \mathrm{id}_A \otimes f} & A \otimes C = \overline{H}_A(C) \\ \downarrow{\scriptstyle \phi_B = \mathrm{in}_A \otimes \mathrm{id}_B} & & \downarrow{\scriptstyle \phi_C = \mathrm{in}_A \otimes \mathrm{id}_C} \\ TB = \overline{H}_\Upsilon(B) = \Upsilon \otimes B & \xrightarrow{\mathrm{id}_\Upsilon \otimes f = Tf} & \Upsilon \otimes C = \overline{H}_\Upsilon(C) = TC \end{array}$$

so that $Tf \circ \Phi_B = (\mathrm{id}_\Upsilon \otimes f) \circ (\mathrm{in}_A \otimes \mathrm{id}_B)$ =(from functorial property of $\otimes$) $= (\mathrm{id}_\Upsilon \circ \mathrm{in}_A) \otimes (f \circ \mathrm{id}_B) = \mathrm{in}_A \otimes f = (\mathrm{in}_A \circ \mathrm{id}_A) \otimes (\mathrm{id}_C \circ f) = (\mathrm{in}_A \otimes \mathrm{id}_C) \circ (\mathrm{id}_A \otimes f) = \phi_C \circ \overline{H}_A(f)$. Consequently, $\phi = \mathrm{id}_{\{_\}} \otimes \mathrm{in}_A$. □

This proposition provides the idea in which way we are able, by using the contravariant functor $\overline{H} : \mathbf{DB}^{\mathrm{OP}} \to \mathbf{DB}^{\mathbf{DB}}$, to define the internalized Yoneda representation of the arrows in $\mathbf{DB}^{\mathrm{OP}}$ by the natural transformations in the category of functors $\mathbf{DB}^{\mathbf{DB}}$: by assuming that in Proposition 43, the natural transformation $\phi = \overline{H}(\mathrm{in}_A^{\mathrm{OP}})$ is just the Yoneda representation of the epimorphism $\mathrm{in}_A^{\mathrm{OP}} : \Upsilon \twoheadrightarrow A$ in $\mathbf{DB}^{\mathrm{OP}}$ (i. e., of the monomorphism $\mathrm{in}_A : A \hookrightarrow \Upsilon$ in **DB**).

Corollary 27. *The internalized Yoneda representation is provided by the contravariant functor* $\overline{H} : \mathbf{DB}^{\mathrm{OP}} \to \mathbf{DB}^{\mathbf{DB}}$*, defined as follows:*

- *For each object A, we define the endofunctor* $\overline{H}_A = \overline{H}(A) : \mathbf{DB} \to \mathbf{DB}$ *such that for any* $f : B \to C$, $\overline{H}_A(f) = \overline{H}(A)(f) = \mathrm{id}_A \otimes f : \overline{H}_A(B) \to \overline{H}_A(C)$, *where* $\overline{H}_A(B) = B \otimes A \simeq A^B$.
- *For each arrow* $g^{\mathrm{OP}} : D \to A$ *in* $\mathbf{DB}^{\mathrm{OP}}$, *we define the natural transformation* $\phi = \overline{H}(g^{\mathrm{OP}}) = g \otimes \mathrm{id}_{\{_\}} : \overline{H}(A) \to \overline{H}(D)$.

In fact, this corollary can be represented by the following commutative diagram in **DB**:

$$\begin{array}{ccc} \overline{H}_A(B) = A \otimes B & \xrightarrow{\overline{H}_A(f) = \mathrm{id}_A \otimes f} & A \otimes C = \overline{H}_A(C) \\ \Big\downarrow{\scriptstyle \phi_B = g \otimes \mathrm{id}_B} & & \Big\downarrow{\scriptstyle \phi_C = g \otimes \mathrm{id}_C} \\ \overline{H}_D(B) = D \otimes B & \xrightarrow{\overline{H}_D(f) = \mathrm{id}_D \otimes f} & D \otimes C = \overline{H}_D(C) \end{array}$$

A Appendix

A.1 Introduction to lattices, algebras and logics

Lattices are the posets (partially ordered sets) such that for all their elements a and b, the set $\{a, b\}$ has both a join (lub – least upper bound) and a meet (glb – greatest lower bound) with a partial order $\leq$ (reflexive, transitive and antisymmetric). A *bounded* lattice has the greatest (top) and least (bottom) element, denoted by convention as 1 and 0. Finite meets in a poset will be written as $1, \wedge$ and finite joins as $0, \vee$. By $(W, \leq, \wedge, \vee, 0, 1)$, we denote a bounded lattice iff for every $a, b, c \in W$ the following equations are valid:

1. $a \vee a = a$, $a \wedge a = a$ (idempotency laws)
2. $a \vee b = b \vee a$, $a \wedge b = b \wedge a$ (commutativity laws)
3. $a \vee (b \vee c) = (a \vee b) \vee c$, $a \wedge (b \wedge c) = (a \vee b) \wedge c$ (associativity laws)
4. $a \vee 0 = a$, $a \wedge 1 = a$
5. $a \vee (b \wedge a) = a$, $a \wedge (b \vee a) = a$ (absorption laws)

It is distributive if it satisfies the distributivity laws.

6. $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$, $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$.

A lattice W is *complete* if each (also infinite) subset $S \subseteq W$ (or, $S \in \mathcal{P}(W)$ where $\mathcal{P}$ is the powerset symbol, with empty set $\emptyset \in \mathcal{P}(W)$) has the least upper bound (lub, supremum) denoted by $\bigvee S \in W$. When S has only two elements, the supremum corresponds to the join operator "$\vee$." Each finite bounded lattice is a complete lattice. Each subset S has the greatest lower bound (glb, infimum) denoted by $\bigwedge S \in W$, given as $\bigvee\{a \in W \mid \forall b \in S . a \leq b\}$. A complete lattice is bounded and has the bottom element $0 = \bigvee \emptyset = \bigwedge W \in W$ and the top element $1 = \bigwedge \emptyset = \bigvee W \in W$. An element $a \in W$ is *compact* iff whenever $\bigwedge S$ exists and $a \leq \bigwedge S$ for $S \subseteq W$; then $a \leq \bigwedge S'$ for some finite $S' \subseteq W$. W is *compactly generated* iff every element in W is a supremum of compact elements. A lattice W is *algebraic* if it is complete and compactly generated.

A function $l : W \to Y$ between the posets W, Y is *monotone* if $a \leq a'$ implies $l(a) \leq l(a')$ for all $a, a' \in W$. Such a function $l : W \to Y$ is said to have a right (or upper) adjoint if there is a function $r : Y \to W$ in the reverse direction such that $l(a) \leq b$ iff $a \leq r(b)$ for all $a \in W, b \in Y$. Such a situation forms a Galois connection and will often be denoted by $l \dashv r$. Then l is called a left (or lover) adjoint of r. If W, Y are complete lattices (posets), then $l : W \to Y$ has a right adjoint iff l preserves all joins (it is *additive*, i. e., $l(a \vee b) = l(a) \vee l(b)$ and $l(0_W) = 0_Y$ where $0_W, 0_Y$ are bottom elements in complete lattices W and Y, respectively). The right adjoint is then $r(b) = \bigvee\{c \in W \mid l(c) \leq b\}$. Similarly, a monotone function $r : Y \to W$ is a right adjoint (it is *multiplicative*, i. e., has a left adjoint) iff r preserves all meets; the left adjoint is then $l(a) = \bigwedge\{c \in Y \mid a \leq r(c)\}$.

Each monotone function $l : W \to Y$ on a complete lattice (poset) W has both a *least* fixed point (Knaster–Tarski) $\mu l \in W$ and *greatest* fixed point $\nu l \in W$. These can be described explicitly as $\mu l = \bigwedge\{a \in W \mid l(a) \leq a\}$ and $\nu l = \bigvee\{a \in W \mid a \leq l(a)\}$.

https://doi.org/10.1515/9783111081670-008

In what follows, we write $b < a$ iff ($b \leq a$ and not $a \leq b$) and we denote by $a \bowtie b$ two unrelated elements in W (so that not ($a \leq b$ or $b \leq a$)). An element in a lattice $c \neq 0$ is a *join-irreducible* element iff $c = a \vee b$ implies $c = a$ or $c = b$ for any $a, b \in W$. An element in a lattice $a \in W$ is an *atom* iff $a > 0$ and $\nexists b.(a > b > 0)$.

A lower set (down closed) is any subset Y of a given poset $(W, \leq)$ such that, for all elements a and b, if $a \leq b$ and $b \in Y$ then $a \in Y$.

A *Heyting algebra* is a distributive bounded lattice W with finite meets and joins such that for each element $a \in W$, the function $(_) \wedge a : W \to W$ has a right adjoint $a \rightharpoonup (_)$ that is also called an algebraic implication. An equivalent definition can be given by considering a bonded distributive lattice such that for all a and b in W there is a greatest element c in W, denoted by $a \rightharpoonup b$, such that $c \wedge a \leq b$, i. e., $a \rightharpoonup b = \bigvee\{c \in W \mid c \wedge a \leq b\}$ (relative pseudo-complement). We say that a lattice is *relatively pseudo-complemented* (r. p. c.) lattice if $a \rightharpoonup b$ exists for every a and b in W. Thus, a Heyting algebra is, by definition, an r. p. c. lattice that has 0.

Formally, a distributive bounded lattice $(W, \leq, \wedge, \vee, 0, 1)$ is a Heyting algebra iff there is a binary operation $\rightharpoonup$ on W such that for every $a, b, c \in W$:

7. $a \rightharpoonup a = 1$,
8. $a \wedge (a \rightharpoonup b) = a \wedge b$, $b \wedge (a \rightharpoonup b) = b$,
9. $a \rightharpoonup (b \wedge c) = (a \rightharpoonup b) \wedge (a \rightharpoonup c)$.

In a Heyting algebra, we can define the negation $\neg a$ as a pseudo-complement $a \rightharpoonup 0$. Then $a \leq \neg\neg a$. Each *finite distributive* lattice is a Heyting algebra. A complete Heyting algebra is a Heyting algebra $\mathbf{H} = (W, \leq, \wedge, \vee, \rightharpoonup, \neg, 0, 1)$, which is complete as a poset. A complete distributive lattice is thus a complete Heyting algebra iff the following *infinite distributivity* holds [113]:

10. $a \wedge \bigvee_{i \in I} b_i = \bigvee_{i \in I}(a \wedge b_i)$, for every $a, b_i \in W$, $i \in I$.

The negation and implication operators can be represented as the following monotone functions: $\neg : W \to W^{\mathrm{OP}}$ and $\rightharpoonup : W \times W^{\mathrm{OP}} \to W^{\mathrm{OP}}$, where W^{OP} is the lattice with inverse partial ordering and $\wedge^{\mathrm{OP}} = \vee$, $\vee^{\mathrm{OP}} = \wedge$.

The following facts are valid in any $\mathbf{H}$:

(H1) $a \leq b$ iff $a \rightharpoonup b = 1$, $(a \rightharpoonup b) \wedge (a \rightharpoonup \neg b) = \neg a$

(H2) $\neg 0 = 0^{\mathrm{OP}} = 1$, $\neg(a \vee b) = \neg a \vee^{\mathrm{OP}} \neg b = \neg a \wedge \neg b$ (additive negation), with the following weakening of classical propositional logic:

(H3) $\neg a \vee b \leq a \rightharpoonup b$, $a \leq \neg\neg a$, $\neg a = \neg\neg\neg a$

(H4) $a \wedge \neg a = 0$, $a \vee \neg a \leq \neg\neg(a \vee \neg a) = 1$ (weakening excluded-middle)

(H5) $\neg a \vee \neg b \leq \neg(a \wedge b)$ (weakening of De Morgan laws).

Notice that since negation $\neg : W \to W^{\mathrm{OP}}$ is a monotonic and additive operator, it is also a *modal* algebraic negation operator. The smallest complete distributive lattice is denoted by $\mathbf{2} = \{0, 1\}$ with twp classic values, false and true, respectively. It is also a complemented Heyting algebra, and hence it is Boolean.

From the point of view of Universal algebra, given a signature Σ with a set of functional symbols $o_i \in \Sigma$ with arity $\mathrm{ar} : \Sigma \to \mathcal{N}$, an algebra (or algebraic structure) $\mathbf{A} = (W, \Sigma_{\mathbf{A}})$ is a carrier set W together with a collection $\Sigma_{\mathbf{A}}$ of operations on W with an arity $n \geq 0$. An n-ary operator (functional symbol) $o_i \in \Sigma$, $\mathrm{ar}(o_i) = n$ on W will be named an n-ary operation (a function) $\widehat{o_i} : W^n \to W$ in $\Sigma_{\mathbf{A}}$ that takes n elements of W and returns a single element of W. Thus, a 0-ary operator (or nullary operation) can be simply represented as an element of W, or a constant, often denoted by a letter like a (thus, all 0-ary operations are included as constants into the carrier set of algebra). An algebra $\mathbf{A}$ is *finite* if the carrier set W is finite; it is *finitary* if each operator in $\Sigma_{\mathbf{A}}$ has a finite arity. For example, a lattice is an algebra with signature $\Sigma_L = \{\wedge, \vee, 0, 1\}$, where $\wedge$ and $\vee$ are binary operations (meet and join operations, respectively), while $0, 1$ are two nullary operators (the constants). The equational semantics of a given algebra is a set of equations E between the terms (or expressions) of this algebra (e. g., a distributive lattice is defined by the first six equations above).

Given two algebras $\mathbf{A} = (W, \Sigma_{\mathbf{A}})$, $\mathbf{A}' = (W', \Sigma_{\mathbf{A}})$ of the same *type* (with the same signature Σ and set of equations), a map $h : W \to W'$ is called a *homomorphism* if for each n-ary operation $\widehat{o_i} \in \Sigma_{\mathbf{A}}$ and $a_1, \ldots, a_n \in W$, $h(\widehat{o_i}(a_1, \ldots, a_n)) = \widehat{o_i'}(h(a_1), \ldots, h(a_n))$. A homomorphism h is called an *isomorphism* if h is a bijection between respective carrier sets; it is called a monomorphism (or embedding) if h is injective function from W into W'. An algebra $\mathbf{A}'$ is called a homomorphic image of $\mathbf{A}$ if there exists a homomorphism from $\mathbf{A}$ onto $\mathbf{A}'$. An algebra $\mathbf{A}'$ is a *subalgebra* of $\mathbf{A}$ if $W' \subseteq W$, the nullary operators are equal and the other operators of $\mathbf{A}'$ are the restrictions of operators of $\mathbf{A}$ to W'.

A *subuniverse* of $\mathbf{A}$ is a subset W' of W, which is closed under operators of $\mathbf{A}$, i. e., for any n-ary operation $\widehat{o}_i \in \Sigma_{\mathbf{A}}$ and $a_1, \ldots, a_n \in W'$, $\widehat{o}_i(a_1, \ldots, a_n) \in W'$. Thus, if $\mathbf{A}$' is a subalgebra of $\mathbf{A}$, then W' is a subuniverse of $\mathbf{A}$. The empty set may be subuniverse, but it is not the underlying carrier set of any subalgebra. If $\mathbf{A}$ has nullary operators (constants), then every subuniverse contains them as well.

Given an algebra $\mathbf{A}$, $\mathrm{Sub}(\mathbf{A})$ denotes the set of subuniverses of $\mathbf{A}$, which is an algebraic lattice. For $Y \subseteq W$, we say that Y *generates* $\mathbf{A}$ (or Y is a set of generators of $\mathbf{A}$) if $W = \mathrm{Sg}(Y) \triangleq \bigcap\{Z \mid Y \subseteq Z \text{ and } Z \text{ is a subuniverse of } \mathbf{A}\}$. Sg is an algebraic closure operator on W: for any $Y \subseteq W$, let $F(Y) = Y \bigcup\{\widehat{o}_i(b_1, \ldots, b_k) | o_i \in \Sigma_{\mathbf{A}} \text{ and } b_1, \ldots, b_k \in Y\}$, with $F^0(Y) = Y$, $F^{n+1}(Y) = F(F^n(Y))$, $n \geq 0$, so that for finitary $\mathbf{A}$, $Y \subseteq F(Y) \subseteq F^2(Y) \subseteq \ldots$, and consequently, $\mathrm{Sg}(Y) = Y \bigcup F(Y) \bigcup F^2(Y) \bigcup \ldots$, and from this it follows that if $a \in \mathrm{Sg}(Y)$ then $a \in F^n(Y)$ for some $n < \omega$; hence, for some *finite* $Z \subseteq Y$, $a \in F^n(Z)$, thus $a \in \mathrm{Sg}(Z)$, i. e., Sg is an algebraic closure operator.

The algebra $\mathbf{A}$ is *finitely generated* if it has a finite set of generators.

Let X be a set of variables. We denote by $\mathcal{T}X$ the set of terms with variables $x_1, x_2, \ldots$ in X of a type Σ of algebras, defined recursively by

– all variables and constants (nullary functional symbols) are in $\mathcal{T}X$;
– if $o_i \in \Sigma$, $n = \mathrm{ar}(o_i) \geq 1$, and $t_1, \ldots, t_n \in \mathcal{T}X$ then $o_i(t_1, \ldots, t_n) \in \mathcal{T}X$.

If $X = \emptyset$, then $\mathcal{T}\emptyset$ denotes the set of ground terms. Given a class K of algebras of the same type (signature Σ), the term algebra $(\mathcal{T}X, \Sigma)$ is a free algebra with universal (initial algebra) property: for every algebra $\mathbf{A} = (W, \Sigma) \in K$ and map $f : X \to W$, there is a unique homomorphism $f_{\#} : \mathcal{T}X \to W$ that extends f to all terms. Given a term $t(x_1, \ldots, x_n)$ over X and given an algebra $\mathbf{A} = (W, \Sigma_{\mathbf{A}})$ of type Σ, we define a mapping $\widehat{t} : W^n \to W$ by: (1) if t is a variable x_i, then $\widehat{t}(a_1, \ldots, a_n) = a_i$ is the i-th projection map; (2) if t is of the form $o_i(t_1(x_1, \ldots, x_n), \ldots, t_k(x_1, \ldots, x_n))$, where $o_i \in \Sigma$, then $\widehat{t}(a_1, \ldots, a_n) = \widehat{o}_i(\widehat{t}_1(a_1, \ldots, a_n), \ldots, \widehat{t}_k(a_1, \ldots, a_n))$. Thus, $\widehat{t}$ is the *term function* on $\mathbf{A}$ corresponding to term t. For any subset $Y \subseteq W$,

$$\mathrm{Sg}(Y) = \{\widehat{t}(a_1, \ldots, a_n) \mid t \text{ is } n\text{-ary term of type } \Sigma, n < \omega \text{ and } a_1, \ldots, a_n \in Y\}.$$

The product of two algebras of the same type $\mathbf{A}$ and $\mathbf{A}'$ is the algebra $\mathbf{A}\times\mathbf{A}' = (W\times W', \Sigma_\times)$ such that for any n-ary operator $\widehat{o_{i,2}} \in \Sigma_\times$ and $(a_1, b_1), \ldots, (a_n, b_n) \in W \times W'$, $n \geq 1$, $\widehat{o_{i,2}}((a_1, b_1), \ldots, (a_n, b_n)) = (\widehat{o_i}(a_1, \ldots, a_n), \widehat{o_i'}(b_1, \ldots, b_n))$. In what follows, where there is no ambiguity, we will write $o_i(a_1, \ldots, a_n)$ for $\widehat{o_i}(a_1, \ldots, a_n)$ as well and Σ for $\Sigma_{\mathbf{A}}$ of any algebra $\mathbf{A}$ of this type Σ.

Given a Σ-algebra $\mathbf{A}$ with the carrier W, we say that an equivalence relation Q on W agrees with the n-ary operation $o_i \in \Sigma$ when for n-tuples $(a_1, \ldots, a_n), (b_1, \ldots, b_n) \in W^n$ we have $(o_i(a_1, \ldots, a_n), o_i(b_1, \ldots, b_n)) \in Q$ if $(a_i, b_i) \in Q$ for $i = 1, \ldots, n$. We say that equivalence relation Q on a Σ-algebra $\mathbf{A}$ is a *congruence* on $\mathbf{A}$ if it agrees with every operation in Σ. If $\mathbf{A}$ is a Σ-algebra and Q a congruence on $\mathbf{A}$, then there exists a unique Σ-algebra on the quotient set W/Q of the carrier W of $\mathbf{A}$ such that the natural mapping $W \to W/Q$ (which maps each element $a \in W$ into its equivalence class $[a] \in W/Q$) is a homomorphism. We will denote such an algebra as $\mathbf{A}/Q = (W/Q, \Sigma)$ and will call it a *quotient algebra* of an algebra $\mathbf{A}$ by the congruence Q, such that for each its k-ary operation $\widehat{o}_i'$, we have $\widehat{o}_i'([a_1], \ldots, [a_k]) = [\widehat{o}_i(a_1, \ldots, a_k)]$.

Let K be a class of algebras of the same type. We say that K is a *variety* if K is closed under homomorphic images, subalgebras and products. Each variety can be seen as a *category* with objects being the algebras and arrows the homomorphisms between them.

The fundamental Birkhoff's theorem in Universal algebra demonstrates that a class of algebras forms a variety iff it is equationally definable. For example, the class of all Heyting algebras (which are definable by the set E of nine equations above), denoted by $\mathcal{HA} = (\Sigma_H, E)$, is a variety. Arend Heyting produced an axiomatic system of propositional logic, which was claimed to generate as theorems precisely those sentences that are valid according to the *intuitionistic* conception of truth. Its axioms are all axioms of the Classic Propositional Logic (CPL) having a set of propositional symbols $p, q, \ldots \in \mathrm{PR}$ and the following axioms (ϕ, ψ, φ denotes arbitrary propositional formulae):

(1) $\phi \Rightarrow (\phi \wedge \phi)$
(2) $(\phi \wedge \psi) \Rightarrow (\psi \wedge \phi)$
(3) $(\phi \Rightarrow \psi) \Rightarrow ((\phi \wedge \varphi) \Rightarrow (\psi \wedge \varphi))$

(4) $((\phi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)) \Rightarrow (\phi \Rightarrow \varphi)$
(5) $\psi \Rightarrow (\phi \Rightarrow \psi)$
(6) $(\phi \wedge (\phi \Rightarrow \psi)) \Rightarrow \psi$
(7) $\phi \Rightarrow (\phi \vee \psi)$
(8) $(\phi \vee \psi) \Rightarrow (\psi \vee \phi)$
(9) $((\phi \Rightarrow \varphi) \wedge (\psi \Rightarrow \varphi)) \Rightarrow ((\phi \vee \psi) \Rightarrow \varphi)$

and for negation in CPL:
(10) $\phi \Rightarrow (\neg\phi \Rightarrow \psi)$
(11) $((\phi \Rightarrow \psi) \wedge (\phi \Rightarrow \neg\psi)) \Rightarrow \neg\phi$
(12) $\phi \vee \neg\phi$

except for the 12th axiom of "excluded middle," which according to a constructivist attitude (L. E. J. Brouwer) has to be replaced by "*either I have constructively demonstrated ϕ, or I have constructively demonstrated that ϕ is false,*" equivalent to modal formula $\Box\phi \vee \Box\neg\phi$, where $\Box$ is a "necessity" universal modal operator in S4 modal logic (with transitive and symmetric accessibility relations between the possible worlds in Kripke semantics, i. e., where this relation is partial ordering $\leq$). In the same constructivist attitude, $\neg\neg\phi \Rightarrow \phi$ is not valid (different from CLP). According to Brouwer, to say that ϕ is not true means only that I have not at this time constructed ϕ, which is not the same as saying ϕ is false.

In fact, in Intuitionistic Logic (IL), $\phi \Rightarrow \psi$ is equivalent to $\Box(\phi \Rightarrow_c \psi)$, i. e., to $\Box(\neg_c\phi \vee \psi)$ where "$\Rightarrow_c$" is a classical logical implication and "$\neg_c$" is a classical negation and $\neg\phi$ is equivalent to $\Box\neg_c\phi$. Thus, in IL, the conjunction and disjunction are that of CPL and only the implication and negation are modal versions of classical versions of the implication and negation, respectively.

A.1.1 Introduction to deductive logic and binary sequent calculus

The concepts introduced here, based on the work of Tarski, are basic to the development of nonmonotonic logic in the rest of the paper.

We assume that a fixed 2-valued (f,false and t,true) object language $\mathcal{L}$ is given. The details of $\mathcal{L}$ are left open, except that it contains the standard connectives, $\Rightarrow, \wedge, \vee$ (material implication, conjunction and disjunction, respectively). Hence, the set $\mathcal{L}_0$ of *sentences* of $\mathcal{L}$ is closed under the rules: $f \in \mathcal{L}_0$; if $\phi, \psi \in \mathcal{L}_0$ then $\phi \Rightarrow \psi, \phi \wedge \psi, \phi \vee \psi \in \mathcal{L}_0$. $\neg\phi$ is taken as a abbreviation of $\phi \Rightarrow f$. We use $\forall$ symbol for "for all" quantifier, and $\exists$ for existential quantifier.

A *consequence relation* is a binary relation $\vdash$, which takes a set of sentences $\Gamma \subseteq \mathcal{L}_0$ as its first argument and a single sentences $\phi \in \mathcal{L}_0$ as its second, denoted by $\Gamma \vdash \phi$. Equivalently, we can define a *consequence operation* (infinitary) mapping $C : \mathcal{P}(\mathcal{L}_0) \to \mathcal{P}(\mathcal{L}_0)$ (here $\mathcal{P}(\mathcal{L}_0)$ denotes the set of *all* subsets of $\mathcal{L}_0$), such that $C(\Gamma) = \{\phi \mid \Gamma \vdash \phi\}$,

and vice versa, $\Gamma \vdash \phi$ iff $\phi \in C(\Gamma)$. The finitary version of this operator is a mapping $C_{\text{fin}} : \mathcal{P}_{\text{fin}}(\mathcal{L}_0) \to \mathcal{P}(\mathcal{L}_0)$, where $\mathcal{P}_{\text{fin}}$ is a finitary powerset operator. We write $C(\Gamma, \Delta)$ instead of $C(\Gamma \bigcup \Delta)$, and $C(\phi)$ instead of $C(\{\phi\})$. The following table defines properties of the *deductive* monotonic consequence relation (reflexivity, cut, monotonicity and compactness):

The finitistic or "Gentzen-style" (Table A.1):

Table A.1: The finitistic or "Gentzen-style"

$\phi \in \Gamma$ implies $\Gamma \vdash \phi$
$\Gamma \bigcup \Delta \vdash \phi$, $\forall \psi \in \Delta.(\Gamma \vdash \psi)$ implies $\Gamma \vdash \phi$
$\Gamma \vdash \phi$, $\Gamma \subseteq \Delta$ implies $\Delta \vdash \phi$
if $\Gamma \vdash \phi$ then for some finite $\Delta \subseteq \Gamma$, $\Delta \vdash \phi$

The infinitistic or "Tarski-style" (Table A.2):

Table A.2: The infinitistic or "Tarski-style"

$\Gamma \subseteq C(\Gamma)$
$\Delta \subseteq C(\Gamma)$ implies $C(\Gamma \bigcup \Delta) \subseteq C(\Gamma)$
$\Gamma \subseteq \Delta$ implies $C(\Gamma) \subseteq C(\Delta)$
$C(\Gamma) \subseteq \bigcup\{C(\Delta) \mid \Delta \subseteq \Gamma$ and Δ is finite$\}$

For example, the classical propositional logic is a deductive logic.

It is easy to verify that C is a closure idempotent operator: for any Γ, we obtain the closet set, called *theory* $T = C(\Gamma) = C(C(\Gamma))$.

For $\Gamma, \Delta \subseteq \mathcal{L}_0$, we shall say that Γ is *consistent* iff $C(\Gamma) \neq \mathcal{L}_0$ and that Γ is *consistent with* Δ iff $C(\Gamma, \Delta) \neq \mathcal{L}_0$. A set is *inconsistent* iff it is not consistent.

A set Γ is $\mathcal{L}$-maximal iff is consistent and for every Δ, if $\Gamma \subseteq \Delta$ and Δ consistent, then $\Gamma = \Delta$.

We denote by $\mathcal{M}_L$ the set of all $\mathcal{L}$-maximal sets, and by $\mathcal{T}_L$ the set of all theories in $\mathcal{P}(\mathcal{L}_0)$, and by $|\Gamma| = \{M \in \mathcal{M}_L \mid \Gamma \subseteq M\}$ the set of all maximal extensions of Γ.

For any deductive logic, the following properties hold:

1. Every $\mathcal{L}$-maximal set Γ is a theory (i. e., $\Gamma = C(\Gamma)$).
2. (Lindenbaum) Every consistent set is included in some $\mathcal{L}$-maximal theory.
3. If $\phi \notin C(\Gamma)$, then there exists a $\mathcal{L}$-maximal theory M such that $C(\Gamma) \subseteq M$ and $\phi \notin M$.
4. $C(\Gamma) = \bigcap |\Gamma|$, i. e., ϕ is a consequence of Γ iff ϕ belongs to every maximal extension of Γ.

So far, our discussion has been purely syntactical and proof-theoretic. We shall also suppose that, with the language $\mathcal{L}$ and the consequence operator C comes a suitable *semantics* in the form of a set $\mathcal{W}$ (the universe), the elements of which we shall call worlds,

and the relation $\models \subseteq \mathcal{W} \times \mathcal{P}(\mathcal{L}_0)$ of *satisfaction* between worlds and formulae: $\phi \in C(\Gamma)$ iff $\forall w \in \mathcal{W}$. ($w \models \Gamma$ implies $w \models \phi$).

We use a deductive logics as underlying logic in order to define the inference operator for (generally nonmonotonic) query-answering in general data integration framework.

Binary sequent calculus

It has been developed by Gentzen [114], inspired on ideas of Paul Hertz [115]. Given a propositional logic language $\mathcal{L}_A$ (a set of logic formulae), a binary sequent is a consequence pair of formulae, $s = (\phi; \psi) \in \mathcal{L}_A \times \mathcal{L}_A$, denoted also by $\phi \vdash \psi$.

A Gentzen system, denoted by a pair $\mathcal{G} = \langle \mathbb{L}, \Vdash \rangle$, where $\Vdash$ is a consequence relation on a set of sequents in $\mathbb{L} \subseteq \mathcal{L}_A \times \mathcal{L}_A$, is said to be *normal* if it satisfies the following conditions: for any sequent $s = \phi \vdash \psi \in \mathbb{L}$ and a set of sequents $\Gamma = \{s_i = \phi_i \vdash \psi_i \in \mathbb{L} \mid i \in I\}$,

1. reflexivity: if $s \in \Gamma$, then $\Gamma \Vdash s$
2. transitivity: if $\Gamma \Vdash s$ and for every $s' \in \Gamma$, $\Theta \Vdash s'$, then $\Theta \Vdash s$
3. finiteness: if $\Gamma \Vdash s$, then there is finite $\Theta \subseteq \Gamma$ such that$\Theta \Vdash s$
4. for any homomorphism σ from $\mathbb{L}$ into itself (i. e., a substitution), if $\Gamma \Vdash s$, then

$$\sigma[\Gamma] \Vdash \sigma(s), \quad \text{i. e., } \{\sigma(\phi_i) \vdash \sigma(\psi_i) \mid i \in I\} \Vdash (\sigma(\phi) \vdash \sigma(\psi)).$$

Notice that from (1) and (2) we obtain this monotonic property:

5. if $\Gamma \Vdash s$ and $\Gamma \subseteq \Theta$, then $\Theta \Vdash s$.

We denote the Tarskian closure operator by $C : \mathcal{P}(\mathbb{L}) \to \mathcal{P}(\mathbb{L})$, such that $C(\Gamma) =_{\text{def}} \{s \in \mathbb{L} \mid \Gamma \Vdash s\}$, with the properties: $\Gamma \subseteq C(\Gamma)$ (from reflexivity (1)); it is monotonic, i. e., $\Gamma \subseteq \Gamma_1$ implies $C(\Gamma) \subseteq C(\Gamma_1)$ (from (5)), and an involution $C(C(\Gamma)) = C(\Gamma)$ as well. Thus, we obtain:

6. $\Gamma \Vdash s$ iff $s \in C(\Gamma)$.

Any sequent theory $\Gamma \subseteq \mathbb{L}$ is said to be a *closed* theory iff $\Gamma = C(\Gamma)$. This closure property corresponds to the fact that $\Gamma \Vdash s$ iff $s \in \Gamma$.

Each sequent theory Γ can be considered as a bivaluation (a characteristic function) $\beta : \mathbb{L} \to \mathbf{2}$ such that for any sequent $s \in \mathbb{L}$, $\beta(s) = 1$ iff $s \in \Gamma$.

A.1.2 Introduction to first-order logic and Tarski's interpretations

We will shortly introduce the syntax of the first-order logic (FOL) language $\mathcal{L}$, and its extensional semantics based on Tarski's interpretations, is as follows.

Definition 55. The syntax of the first-order logic language $\mathcal{L}$ is as follows:

Logic operators $(\wedge, \neg, \exists)$ over bounded lattice of truth values $\mathbf{2} = \{f, t\}, f$ for falsity and t for truth; Predicate letters $p_1^{k_1}, p_2^{k_2}, \ldots$ with a given arity $k_i \geq 1$, $i = 1, 2, \ldots$, and special built-in identity binary predicate "$\doteq$," in P; Functional letters $f_1^{k_1}, f_2^{k_2}, \ldots$ with a given arity $k_i \geq 1$ in F (language constants $c, d, \ldots$ are considered as particular case of nullary functional letters); Variables $x, y, z, \ldots$ in $\mathcal{V}$, and punctuation symbols (comma, parenthesis).

With the following simultaneous inductive definition of *term* and *formula*:

1. All variables and constants are terms.
2. If $t_1, \ldots, t_k$ are terms and $f_i^k \in F$ is a k-ary functional symbol, then $f_i^k(t_1, \ldots, t_k)$ is a term, while $p_i^k(t_1, \ldots, t_k)$ is a formula for a k-ary predicate letter $p_i^k \in P$.
3. If ϕ and ψ are formulae, then $(\phi \wedge \psi)$, $\neg\phi$, and $(\exists x_i)\phi$ for $x_i \in \mathcal{V}$ are formulae.

An interpretation (Tarski) I_T consists in a nonempty domain $\mathcal{D}$ and a mapping that assigns to any predicate letter $p_i^k \in P$ a relation $R = I_T(p_i^k) \subseteq \mathcal{D}^k$, to any functional letter $f_i^k \in F$ a function $I_T(f_i^k) : \mathcal{D}^k \to \mathcal{D}$, or equivalently, its graph relation $R = I_T(f_i^k) \subseteq \mathcal{D}^{k+1}$ where the $k+1$-th column is the resulting function's value, and to each individual constant $c \in F$ one given element $I_T(c) \in \mathcal{D}$.

A predicate logic $\mathcal{L}_P$ is a subset of the FOL without the quantifier $\exists$.

Remark. The *propositional* logic can be considered as a particular case of predicate logic when all symbols in P are nullary, i. e., a set of propositional symbols, while F, $\mathcal{D}$ and $\mathcal{V}$ are empty sets. By considering that $\mathcal{D}^0 = \{\langle\rangle\}$ is a singleton set composed by empty tuple $\langle\rangle$, then for any $p_i \in P$, $I_T(p_i) \subseteq \mathcal{D}^0$, i. e., $I_T(p_i) = f$ (empty set) or $I_T(p_i) = t$ (singleton set $\{\langle\rangle\}$). That is, I_T becomes an interpretation $I_T : P \to \mathbf{2}$ of this logic, which can be homomorphically extended to all formulae in the unique standard way. □

In a formula $(\exists x)\phi$, the formula ϕ is called an "action field" for the quantifier $(\exists x)$. A variable y in a formula ψ is called a bounded variable iff it is the variable of a quantifier $(\exists y)$ in ψ, or it is in the action field of a quantifier $(\exists y)$ in the formula ψ. A variable x is free in ψ if it is not bounded.

The universal quantifier is defined by $\forall = \neg\exists\neg$. Disjunction and implication are expressed by $\phi \vee \psi = \neg(\neg\phi \wedge \neg\psi)$, and $\phi \Rightarrow \psi = \neg\phi \vee \psi$. In the FOL with the identity $\doteq$, the formula $(\exists_1 x)\phi(x)$ denotes the formula $(\exists x)\phi(x) \wedge (\forall x)(\forall y)(\phi(x) \wedge \phi(y) \Rightarrow (x \doteq y))$.

We can introduce the sorts in order to be able to assign each variable x_i to a sort $S_i \subseteq \mathcal{D}$ where $\mathcal{D}$ is a given domain for the FOL (e. g., for natural numbers, for reals, for dates, etc. as used for some attributes in database relations). An assignment $g : \mathcal{V} \to \mathcal{D}$ for variables in $\mathcal{V}$ is applied only to free variables in terms and formulae. If we use sorts for variables, then for each sorted variable $x_i \in \mathcal{V}$ an assignment g must satisfy the auxiliary condition $g(x_i) \in S_i$.

Such an assignment $g \in \mathcal{D}^{\mathcal{V}}$ can be recursively and uniquely extended into the assignment $g^* : \mathcal{T} \to \mathcal{D}$, where $\mathcal{T}$ denotes the set of all terms, by:

1. $g^*(t_i) = g(x) \in \mathcal{D}$ if the term t_i is a variable $x \in \mathcal{V}$.

2. $g^*(t_i) = I_T(c) \in \mathcal{D}$ if the term t_i is a constant $c \in F$.
3. If a term t_i is $f_i^k(t_1, \ldots, t_k)$, where $f_i^k \in F$ is a k-ary functional symbol and $t_1, \ldots, t_k$ are terms, then $g^*(f_i^k(t_1, \ldots, t_k)) = I_T(f_i^k)(g^*(t_1), \ldots, g^*(t_k))$ or, equivalently, in the graph-interpretation of the function, $g^*(f_i^k(t_1, \ldots, t_k)) = u$ such that $(g^*(t_1), \ldots, g^*(t_k), u) \in I_T(f_i^k) \subseteq \mathcal{D}^{k+1}$.

In what follows, we will use the graph interpretation for functions in FOL. We denote by t_i/g (or ϕ/g) the ground term (or formula) without free variables, obtained by assignment g from a term t_i (or a formula ϕ), and by $\phi[x/t_i]$ the formula obtained by uniformly replacing x by a term t_i in ϕ. A *sentence* is a formula having no free variables.

A Herbrand base of a logic $\mathcal{L}$ is defined by

$$H = \{p_i^k(t_1, \ldots, t_k) \mid p_i^k \in P \text{ and } t_1, \ldots, t_k \text{ are ground terms}\}.$$

A *FOL interpretation* $\mathfrak{I}$ is a triple $(\mathcal{D}, g, I_T)$, where $\mathcal{D}$ is domain of interpretation, $g \in \mathcal{D}^{\mathcal{V}}$ an assignment and I_T is a function (called *valuation* v as well) that maps constants of FOL (nullary functions) into domain $\mathcal{D}$, the k-ary predicate into relation $\mathcal{D}^k$ and k-ary function into relation $\mathcal{D}^{k+1}$ as explained above.

Tarski defined the satisfaction for the logic formulae in $\mathcal{L}$ and a given assignment $g : \mathcal{V} \to \mathcal{D}$ inductively (recursively), as follows:

If a formula ϕ is an atomic formula $p_i^k(t_1, \ldots, t_k)$, then this assignment g satisfies ϕ iff $(g^*(t_1), \ldots, g^*(t_k)) \in I_T(p_i^k)$; g satisfies $\neg\phi$ iff it does not satisfy ϕ; g satisfies $\phi \wedge \psi$ iff g satisfies ϕ and g satisfies ψ; g satisfies $(\exists x_i)\phi$ iff exists an assignment $g' \in \mathcal{D}^{\mathcal{V}}$ that may differ from g only for the variable $x_i \in \mathcal{V}$, and g' satisfies ϕ.

A model is $\mathcal{M} = (\mathcal{D}, I_T)$, so that for $g \in \mathcal{D}^{\mathcal{V}}$ and a formula ϕ we use the notation

$$\mathcal{M} \models_g \phi \quad \text{or} \quad I_T^*(\phi/g) = t \in \mathbf{2} \tag{A.1}$$

which means "*the formula ϕ is true in the model $\mathcal{M} = (\mathcal{D}, I_T)$, for assignment g.*"

Remark. In what follows, instead of notation $\mathcal{M} \models_g \phi$ we prefer to use the *nonstandard* functional notation $I_T^*(\phi/g)$, because the new symbol I_T^* denotes a *homomorphism*, introduced in [7], between syntax algebra of FOL and extensional algebra of relations (of *extensional logic),* which allows evaluation of the truth value of a sentence. So, for ground atoms I_T^* is equal to Herbrand interpretation, instead, for atoms with variables only it is equal to original Tarski's mapping I_T. That is, for any atom $p_j^k(\mathbf{x})$ with variables $\mathbf{x} = (x_1, \ldots, x_k)$,

$$I_T(p_j^k) = I_T^*(p_j^k(\mathbf{x})) = \{(g(x_1), \ldots, g(x_k)) \mid g \in \mathcal{D}^{\mathcal{V}}, \text{ and } I_T^*(p_j^k(\mathbf{x})/g) = t\}. \tag{A.2}$$

□

A formula ϕ is `true` for a given interpretation I_T iff ϕ is satisfied by every assignment $g \in \mathcal{D}^{\mathcal{V}}$. A formula ϕ is `valid` (i. e., tautology) iff ϕ is true for every Tarski's interpretation

$I_T \in \mathfrak{I}_T$. An interpretation I_T is a `model` of a set of formulae Γ iff every formula $\phi \in \Gamma$ is true in this interpretation. By FOL(Γ), we denote the FOL with a set of assumptions Γ, and by $\mathfrak{I}_T(\Gamma)$ the subset of Tarski's interpretations that are models of Γ, with $\mathfrak{I}_T(\emptyset) = \mathfrak{I}_T$. A formula ϕ is said to be a *logical consequence* of Γ, denoted by $\Gamma \Vdash \phi$, iff ϕ is true in all interpretations in $\mathfrak{I}_T(\Gamma)$.

Thus, $\Vdash \phi$ iff ϕ is a tautology.

The basic set of axioms of the FOL are that of the propositional logic with two additional axioms:

(A1) $(\forall x)(\phi \Rightarrow \psi) \Rightarrow (\phi \Rightarrow (\forall x)\psi)$, ($x$ does not occur in ϕ and it is not bound in ψ).

(A2) $(\forall x)\phi \Rightarrow \phi[x/t_i]$, (neither x nor any variable in t_i occurs bound in ϕ).

(A3) For the FOL with identity (the binary predicate $\doteq$), we need the *proper* axiom $x_1 \doteq x_2 \Rightarrow (x_1 \doteq x_3 \Rightarrow x_2 \doteq x_3)$.

We denote by $R_=$ the Tarski's interpretation of $\doteq$.

The inference rules are Modus Ponens and generalization (G) "*if ϕ is a theorem and x is not bound in ϕ, then $(\forall x)\phi$ is a theorem.*"

Here, the FOL denotes this minimal first-order logic (consistent and complete), while FOL(Γ) denotes a logic extended by a new set of *proper* axioms in Γ, and generally (by Gödel theorem), it is not complete.

In what follows any open sentence, a formula $\phi(\mathbf{x})$ with a nonempty tuple of free variables $\mathbf{x} = (x_1, \ldots, x_m)$, will be called a *m*-ary *virtual predicate*, denoted also by $\phi(x_1, \ldots, x_m)$. This definition contains the precise method of establishing the *ordering* of variables in this tuple: such a method that will be adopted here is the ordering of appearance, from left to right, of free variables in ϕ.

The FOL is considered as an extensional logic because two open sentences with the same tuple of variables $\phi(x_1, \ldots, x_m)$ and $\psi(x_1, \ldots, x_m)$ are `equal` iff they have the *same extension* in a given interpretation I_T, i. e., iff $I_T^*(\phi(x_1, \ldots, x_m)) = I_T^*(\psi(x_1, \ldots, x_m))$, where I_T^* is the unique extension of I_T to all formulae, as follows:

1. For a (closed) sentence ϕ/g, we have that $I_T^*(\phi/g) = t$ iff g satisfies ϕ, as recursively defined above.
2. For an open sentence ϕ with the tuple of free variables $(x_1, \ldots, x_m)$, we have that $I_T^*(\phi(x_1, \ldots, x_m)) \triangleq \{(g(x_1), \ldots, g(x_m)) \mid g \in \mathcal{D}^{\mathcal{V}} \text{ and } I_T^*(\phi/g) = t\}$.

It is easy to verify that for a formula ϕ with the tuple of free variables $(x_1, \ldots, x_m)$,

$$I_T^*(\phi(x_1, \ldots, x_m)/g) = t \quad \text{iff} \quad (g(x_1), \ldots, g(x_m)) \in I_T^*(\phi(x_1, \ldots, x_m)).$$

This extensional *equality* of virtual predicates can be generalized to the extensional *equivalence* when both predicates ϕ, ψ has the same set of free variables but their ordering in the *tuples* of free variables are not identical: such two virtual predicates are equivalent if the extension of the first is equal to the proper permutation of columns of

the extension of the second virtual predicate. It is easy to verify that such an extensional equivalence corresponds to the logical equivalence denoted by $\phi \equiv \psi$.

Let $\Re = \bigcup_{k\in\mathbb{N}} \mathcal{P}(\mathcal{D}^k) = \sum_{k\in\mathbb{N}} \mathcal{P}(D^k)$ be the set of all k-ary relations over a domain $\mathcal{D}$, where $k \in \mathbb{N} = \{0, 1, 2, \ldots\}$. Then this extensional equivalence between two relations $R_1, R_2 \in \Re$ with the same arity will be denoted by $R_1 \approx R_2$, while the extensional identity will be denoted in the standard way by $R_1 = R_2$.

A.1.3 Introduction to multimodal logics and Kripke semantics

A multimodal logic is a standard predicate/propositional logic (see Definition 55) extended by a number of *existential modal* operators $\Diamond_i, i \geq 1$. In the standard Kripke semantics, each existential modal operator $\Diamond_i$ is defined by an accessibility binary relation $\mathcal{R}_i \subseteq \mathcal{W} \times \mathcal{W}$, for a given set of *possible worlds* $\mathcal{W}$. A more exhaustive and formal introduction to modal logics and their Kripke's interpretations can easily be found in the literature, e. g., in [116].

Here, only a short version will be given, in order to clarify the definitions used in the next paragraphs. We define $\mathcal{N} = \{0, 1, 2, \ldots, n\} \subset \mathbb{N}$ where n is a maximal arity of symbols in the finite set $P \bigcup F$ of predicate and functional symbols respectively.

In the case of the propositional logics, we have that $n = 0$, so that P is a set of propositional symbols (that are the nullary predicate symbols) and $F = \emptyset$ is the empty set. Here, we will present two definitions for modal logics, one for the propositional and other for predicate logics, as is used in current literature.

Definition 56 (Propositional multimodal logic). By $\mathcal{M} = (\mathcal{W}, \{\mathcal{R}_i\}, I_K)$, we denote a multimodal Kripke's interpretation with a set of possible worlds $\mathcal{W}$, the accessibility relations $\mathcal{R}_i \subseteq \mathcal{W} \times \mathcal{W}$, $i = 1, 2, \ldots$ and a mapping $I_K : P \to \mathbf{2}^{\mathcal{W}}$, such that for any propositional letter $p_i \in P$, the function $I_K(p_i) : \mathcal{W} \to \mathbf{2}$ defines the truth of p_i in a world $w \in \mathcal{W}$.

For any formula φ, we define $\mathcal{M} \vDash_w \varphi$ iff φ is satisfied in a world $w \in \mathcal{W}$. For example, a given propositional letter p_i is true in w, i. e., $\mathcal{M} \vDash_w p_i$, iff $I_K(p_i)(w) = t$.

The Kripke semantics is extended to all formulae as follows:

$$\begin{aligned}
&\mathcal{M} \vDash_w \varphi \wedge \phi \quad \text{iff} \quad \mathcal{M} \vDash_w \varphi \text{ and } \mathcal{M} \vDash_w \phi,\\
&\mathcal{M} \vDash_w \neg\varphi \quad \text{iff not} \quad \mathcal{M} \vDash_w \varphi,\\
&\mathcal{M} \vDash_w \Diamond_i\varphi \quad \text{iff exists} \quad w' \in \mathcal{W} \text{ such that } (w, w') \in \mathcal{R}_i \text{ and } \mathcal{M} \vDash_{w'} \varphi.
\end{aligned}$$

The universal modal operator $\Box_i$ is equal to $\neg\Diamond_i\neg$. A formula φ is said to be *true in a Kripke's interpretation* $\mathcal{M}$ if for each possible world w, $\mathcal{M} \vDash_w \varphi$. A formula is said to be *valid* if it is true in each interpretation.

In a more general case, when $n \geq 1$ we have the multimodal predicate logics defined as follows.

Definition 57 (Predicate multimodal logic). By $\mathcal{M} = (\mathcal{W}, \{\mathcal{R}_i \mid 1 \leq i \leq k\}, \mathcal{D}, I_K)$, we denote a multimodal Kripke model with finite $k \geq 1$ modal operators with a set of possible worlds $\mathcal{W}$, the accessibility relations $\mathcal{R}_i \subseteq \mathcal{W} \times \mathcal{W}$, nonempty domain $\mathcal{D}$ and a mapping $I_K : \mathcal{W} \times (P \bigcup F) \to \bigcup_{n \in \mathcal{N}} (\mathbf{2} \bigcup \mathcal{D})^{\mathcal{D}^n}$, such that for any world $w \in \mathcal{W}$,

1. For any functional letter $f_i^k \in F$, $I_K(w, f_i^k) : \mathcal{D}^k \to \mathcal{D}$ is a function (interpretation of f_i^k in w).
2. For any predicate letter $p_i^k \in P$, the function $I_K(w, p_i^k) : \mathcal{D}^k \to \mathbf{2}$ defines the extension of p_i^k in a world w,

$$\|p_i^k(x_1, \ldots, x_k)\|_{\mathcal{M},w} \triangleq \{(d_1, \ldots, d_k) \in \mathcal{D}^k \mid I_K(w, p_i^k)(d_1, \ldots, d_k) = t\}.$$

For any formula φ, we define $\mathcal{M} \vDash_{w,g} \varphi$ iff φ is satisfied in a world $w \in \mathcal{W}$ for a given assignment $g : \mathcal{V} \to \mathcal{D}$. For example, a given atom $p_i^k(x_1, \ldots, x_k)$ is satisfied in w by assignment g, i. e., $\mathcal{M} \vDash_{w,g} p_i^k(x_1, \ldots, x_k)$, iff $I_K(w, p_i^k)(g(x_1), \ldots, g(x_k)) = t$.

The Kripke semantics is extended to all formulae as follows:

$$\begin{aligned}
&\mathcal{M} \vDash_{w,g} \varphi \wedge \phi \quad \text{iff} \quad \mathcal{M} \vDash_{w,g} \varphi \text{ and } \mathcal{M} \vDash_{w,g} \phi,\\
&\mathcal{M} \vDash_{w,g} \neg\varphi \quad \text{iff not} \quad \mathcal{M} \vDash_{w,g} \varphi,\\
&\mathcal{M} \vDash_{w,g} \Diamond_i\varphi \quad \text{iff exists} \quad w' \in \mathcal{W} \text{ such that } (w, w') \in \mathcal{R}_i \text{ and } \mathcal{M} \vDash_{w',g} \varphi.
\end{aligned}$$

A formula φ is said to be *true in a Kripke's interpretation* $\mathcal{M}$ if for each assignment function g and possible world w, $\mathcal{M} \vDash_{w,g} \varphi$. A formula is said to be *valid* if it is true in each interpretation.

Any virtual predicate $\phi(x_1, \ldots, x_k)$ has different extensions $\|\phi(x_1, \ldots, x_k)\|_{\mathcal{M},w} \triangleq \{(g(x_1), \ldots, g(x_k)) \mid g \in \mathcal{D}^{\mathcal{V}} \text{ and } \mathcal{M} \vDash_{w,g} \phi\}$ for different possible worlds $w \in \mathcal{W}$. Thus, we cannot establish the simple extensional identity for two concepts as in FOL.

Remark. We can consider the following reductions of the modal predicate logics.

The propositional modal logic can be considered as a particular case of a predicate modal logic when all symbols in P are nullary, i. e., a set of propositional symbols, while F, $\mathcal{D}$ and $\mathcal{V}$ are empty sets. So that I_K is reduced to a mapping $I_K : P \to \mathbf{2}^{\mathcal{W}}$, such that we do not use the assignments g, and for any propositional letter $p_i \in P$, the function $I_K(p_i) : \mathcal{W} \to \mathbf{2}$ defines the truth of p_i in a world $w \in \mathcal{W}$.

A.2 Basic category theory

Category theory is an area of mathematics that examines in an abstract way the properties of particular mathematical concepts. Many significant areas of mathematics can be formalized as categories, and the use of category theory allows many intricate and subtle mathematical results in these fields to be stated, and proved, in a much simpler way than without the use of categories.

The most accessible example of a category is the category **Set** of (small) sets, where the objects are sets and the arrows are functions between them. However, it is important to note that the objects of a category need not be sets or the arrows functions; any way of formalizing a mathematical concept such that it meets the basic conditions on the behavior of objects and arrows is a valid category, and all the results of category theory will apply to it.

Category theory introduces a new, original linguistic component in mathematics: the language of arrows, commutative diagrams and, in the case of monoidal categories and in the higher-dimensional case, of geometric transformations of diagrams. Although these were for a long time considered to be merely linguistic devices, they should be seen as genuine conceptual ingredients of mathematical knowledge, which capture conceptual dependencies.

Categories were first introduced by Samuel Eilenberg and Saunders Mac Lane in 1942–1945 [12], in connection with algebraic topology. The study of categories is an attempt to axiomatically capture what is commonly found in various classes of related mathematical structures by relating them to the structure-preserving functions between them. A systematic study of category theory then allows us to prove general results about any of these types of mathematical structures from the axioms of a category.

In 1930s, the development and status of (First-order Logic) FOL is intimately linked to the formalization of set theory and the crystallization of foundational studies in the framework of ZF set theory. It is not entirely absurd to affirm that FOL and ZF were two sides of the same constitutive movement. FOL was enough to formalize set theories like ZF and the latter was enough to express and prove basic properties of FOL.The logicians of that period assimilated the axiomatization of set theory with the foundations of abstract mathematics. This assimilation probably made perfect sense in the 1930s and that many mathematicians identified abstract or modern mathematics with the set-based mathematics developed at that time. There was, however, a hiatus introduced between the practice of abstract mathematics and the foundational research. In some sense, the latter still had deep roots in nineteenth century mathematics via the works of Frege, Peano, Russell and Whitehead. This divergence increased considerably with the introduction of categorical methods in algebraic geometry, homological algebra, algebraic topology and homotopy theory in the 1960s, allowing various mathematicians to realize, with hindsight, that set theoretical foundations were not faithful to the abstract character of modern mathematics [117]:

> "*In modern times it has been recognized that is possible to elaborate full mathematical theories dealing with elements of which the nature is not specified, that is, with abstract elements. A collection of these elements will be called an abstract set.*" (Fréchet, 1951, p. 147)

The Elementary Theory of the Category of Sets (ETCS) [118] was introduced by Lawvere as early as 1964. In this paper, Lawvere proposed an axiomatization of set theory in a categorical framework. The axiomatization was later subsumed under the notion of an

elementary topos where it became a special type of topos. It is developed in the standard FOL without dependent sorts.[1] Another framework in contrast with the previous approach, Makkai's approach [119] is based on a general purpose formal system, namely First-Order Logic with Dependent Sorts (FOLDS). Homotopy type theory is a recent proposal undergoing a rapid and vigorous development. It is based on Martin-Löf type theory, introduced in the early 1970s to provide a formal framework for constructive mathematics. Homotopy type theory is a genuine type theory, not a first-order logic. Originally, all the variations on Martin-Löf type theory were motivated by the well-known Curry–Howard correspondence, i. e., the propositions-as-types interpretation. The correspondence is given by the following simple rule: a term x of a type X can be thought of as a proof of the corresponding proposition.

Thus, an *abstract set*, different from extensionally defined ZF theory of sets, is a set in which the elements are abstract. This is taken to mean that the nature of the elements is not specified. In other words, the elements have no other properties than those that they have in virtue of being elements of that specific set. The sets of ZF are not abstract in that sense. The abstract notion of set was explicitly endorsed by Lawvere in the 1970s [120]:

> "*These considerations lead one to formulate the following "purified" concept of (constant) abstract set as the one actually used in naive set theoretic practice of modern mathematics: An abstract set X has elements each of which has no internal structure whatsoever; X has no internal structure except for equality and inequality of pairs of elements, and has no external properties save its cardinality: still an abstract set is more refined (less abstract) than a cardinal number in that it does have elements while a cardinal number does not.*" (Lawvere, 1976, p. 119)

That is, nothing is known about the elements of a set X except that there is an internal identity relation between the elements. On the standard, extensional, conception of a set, one looks at the elements of the sets. Since the latter are fully identifiable individuals, they can be known, and knowing a set amounts to knowing these elements. Here, we take a different route: to know a set and its components, including its elements since these become a special case of the more general notion of component, one looks at it "geometrically," i. e., by considering morphisms going in the set and morphisms going out of the set. How do we express identity of abstract sets? It is not a relation, but rather a structure. What can be declared in the given language is that abstract sets can be *isomorphic*. But we do not identify isomorphic sets, unless it is useful for some specific

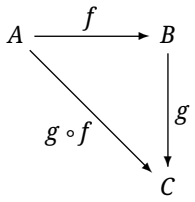
1 One starts with the (first-order) axioms for an elementary topos and add axioms to get as close as possible. to ZFC. We can add to the axioms of an elementary topos the existence of a Natural Number Object (NNO), that the terminal object 1 is a generator and that every epimorphism splits (an equivalent version of the axiom of choice in the context of toposes). It is well known that such toposes are necessarily Boolean. It can be shown that such a topos is equivalent to Bounded Zermelo set theory with choice (BZC). As such, the axiom of replacement is absent. But there are natural ways to add it to the theory. Thus, one gets a somewhat weaker theory than the usual ZFC.

mathematical purpose. Notice that the universe of abstract sets is not a set nor a class: it is a category.

A category **C** consists of the following three mathematical entities: All we had to start with were the notions of (abstract) collection and operation.

- A collection $\mathrm{Ob}_{\mathbf{C}}$ whose elements are called objects.
- A collection $\mathrm{Mor}_{\mathbf{C}}$ whose elements are called morphisms or maps or arrows. Each morphism f has a unique source object A and target object B. The expression (arrow) $f : A \to B$ would be verbally stated as "f is a morphism from A to B." The expression $\hom(A,B)$ — alternatively expressed as $\hom_{\mathbf{C}}(A,B)$, or $\mathrm{mor}(A,B)$, or $\mathbf{C}(A,B)$ — denotes the hom-class of all morphisms from A to B. If for all A, B this hom-class is a set, this category **C** is called *locally small*.
- A binary operation $\circ$, called composition of morphisms, such that for any three objects A, B and C, we have a function $\circ : \hom(A,B) \times \hom(B,C) \to \hom(A,C)$. Each morphism $f \in \hom(A,B)$ has a domain $A = \mathrm{dom}(f)$ and codomain $B = \mathrm{cod}(f)$ which are objects.
  The composition of $f : A \to B$ and $g : B \to C$ is written as $g \circ f$, with two axioms:
  Associativity: If $f : A \to B$, $g : B \to C$ and $h : C \to D$, then $h \circ (g \circ f) = (h \circ g) \circ f$,
  Identity: For every object A, there exists a morphism $\mathrm{id}_A : A \to A$ called the identity morphism for A, such that for every morphism $f : A \to B$, we have $\mathrm{id}_B \circ f = f \circ \mathrm{id}_A = f$.

From these axioms, it can be proved that there is exactly one identity morphism for every object. Because of that we can identify each object with its identity morphism. Relations among morphisms are often depicted using commutative diagrams, with nodes representing objects and arrows representing morphisms, e. g., $g \circ f = h$ is graphically represented by the following commutative diagram:



Examples:

1. Category **Set**, where the objects are *small* sets and the arrows are functions from one set to another.
2. A preorder is a set X together with a binary relation $\leq$ which is reflexive (i. e., $x \leq x$ for all $x \in X$), and transitive (i. e., $x \leq y$ and $y \leq z$ imply $x \leq z$ for all $x, y, z \in X$). This can be seen as a category, with set of objects X and for every pair of objects (x, y) such that $x \leq y$, exactly one arrow $x \to y$.
3. Any set X can be seen as a discrete category, with a set of objects X and only with the identity morphisms.

A morphisms $f : A \to B$ can have any of the following properties:

1. *Monomorphism* (or monic) if $f \circ g_1 = f \circ g_2$ implies $g_1 = g_2$ for all morphisms $g_1, g_2 : X \to A$. It is denoted by $f : A \hookrightarrow B$. This monic is split if there exists $g : B \to A$ such that $g \circ f = \mathrm{id}_A$.
2. *Epimorphism* (or epic) if $g_1 \circ f = g_2 \circ f$ implies $g_1 = g_2$ for all morphisms $g_1, g_2 : B \to X$. It is denoted by $f : A \twoheadrightarrow B$. This epic is called *split* if there exists $g : B \to A$ such that $f \circ g = \mathrm{id}_B$ (then f is a *retraction* of g).
3. *Isomorphism* if there exists a morphism $g : B \to A$ such that $f \circ g = \mathrm{id}_B$ and $g \circ f = \mathrm{id}_A$. Any isomorphism is monic and epic. It is denoted by $f : A \simeq B$. Often we denote simply by $A \simeq B$ the isomorphism of two objects. If f is split epic and monic, f is isomorphism.

For example, in **Set** the epimorphisms are surjective functions, the monomorphisms are injective functions, the isomorphisms are bijective functions.

Duality principle

The definition of epic is *dual* to the definition of monic. Given a category $\mathbf{C} = (\mathrm{Ob}_C, \mathrm{Mor}_C, \mathrm{id}, \circ)$, by $\mathbf{C}^{\mathrm{OP}} = (\mathrm{Ob}_C, \mathrm{Mor}_{C^{\mathrm{OP}}}, \mathrm{id}, \circ^{C^{\mathrm{OP}}})$, we denote the opposite category that have the same objects and where all arrows are reversed: $f^{C^{\mathrm{OP}}} \circ^{C^{\mathrm{OP}}} g^{C^{\mathrm{OP}}} = (g \circ f)^{C^{\mathrm{OP}}}$ (see diagram (A.13)), with bijection $\mathbf{C}(A, B) \simeq \mathbf{C}^{\mathrm{OP}}(B, A)$. Thus, f is monic in the category $\mathbf{C}$ iff f^{OP} is epic in $\mathbf{C}^{\mathrm{OP}}$, and vice versa. In general, given a property P of an object, arrow, diagram, etc., we can associate with P the dual property P^{OP}. The object or arrow has a property P in $\mathbf{C}$ iff it has P^{OP} in $\mathbf{C}^{\mathrm{OP}}$. For example, if $g \circ f$ is monic, then f is monic. From this, by duality, if $f^{\mathrm{OP}} \circ g^{\mathrm{OP}}$ is epic then f^{OP} is epic.

An object X is called *terminal* if for any object Y there is exactly one morphism from it into X in the category. Dually, one object X is called *initial* if for any object Y there is exactly one morphism from X to Y in the category. An object X is called *zero* object if it is both terminal and initial. For example, in **Set** the empty set is initial, while any singleton set is (up to isomorphism) terminal object.

Products of categories

For a given two categories $\mathbf{A}$ and $\mathbf{B}$, we can construct a new category $\mathbf{A} \times \mathbf{B}$, called a product of $\mathbf{A}$ and $\mathbf{B}$, as follows: its object (a, b) is a pair of an object $a \in \mathrm{Ob}_{\mathbf{A}}$ and an object $b \in \mathrm{Ob}_{\mathbf{B}}$, while its arrow $(f, g) : (a, b) \to (c, d)$ is a pair of an arrow $f : a \to c$ in $\mathrm{Mor}_{\mathbf{A}}$ and an arrow $g : b \to d$ in $\mathrm{Mor}_{\mathbf{B}}$. So, the composition of arrows in $\mathbf{A} \times \mathbf{B}$ $\mathbf{A} \times \mathbf{B}$ follows the composition of arrows in $\mathbf{A}$ and in $\mathbf{B}$, relatively. The identity arrows in $\mathbf{A} \times \mathbf{B}$ are pairs of the identity arrows in $\mathbf{A}$ and in $\mathbf{B}$, relatively, i. e., for a given object (a, b) its identity arrow is $\mathrm{id}_{(a,b)} = (\mathrm{id}_a, \mathrm{id}_b)$. We can construct analogously any finite product of a given category $\mathbf{C}$, for $n \geq 1$, denoted by

$$\mathbf{C}^n \triangleq \overbrace{\mathbf{C} \times \cdots \times \mathbf{C}}^{n}$$

such that $\mathbf{C}^1 = \mathbf{C}$, with its n-ary objects and arrows

$$(c_1, \ldots, c_n) \in \mathrm{Ob}_{\mathbf{C}^n} \quad \text{and} \quad (f_1, \ldots, f_n) \in \mathrm{Mor}_{\mathbf{C}^n} \tag{A.3}$$

where for each $1 \leq i \leq n$, $c_i \in \mathrm{Ob}_{\mathbf{C}}$ and $f_i \in \mathrm{Mor}_{\mathbf{C}}$.

Functors are structure-preserving maps between categories. They can be thought of as morphisms in the category **Cat** of all (small) categories as objects, and all functors as morphisms. A (covariant) functor F from a category $\mathbf{C}$ to a category $\mathbf{D}$, written $F : \mathbf{C} \to \mathbf{D}$, consists of a pair of functions $F = (F^0, F^1)$, $F^0 : \mathrm{Ob}_{\mathbf{C}} \to \mathrm{Ob}_{\mathbf{D}}$ and $F^1 : \mathrm{Mor}_{\mathbf{C}} \to \mathrm{Mor}_{\mathbf{D}}$ (in what follows, we will simply use F for both functions as well):

1. For each object X in $\mathbf{C}$, an object $F^0(X)$ in $\mathbf{D}$; and
2. For each morphism $f : X \to Y$ in $\mathbf{C}$, a morphism $F^1(f) : F^0(X) \to F^0(Y)$,

such that the following two properties hold:

3. For every object X in $\mathbf{C}$, $F^1(\mathrm{id}_X) = \mathrm{id}_{F(X)}$;
4. For all morphisms $f : X \to Y$ and $g : Y \to Z$, $F^1(g \circ f) = F^1(g) \circ F^1(f)$.

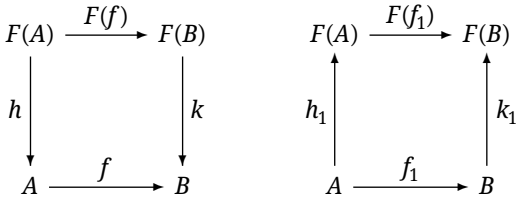
A contravariant functor $F : \mathbf{C} \to \mathbf{D}$, is like a covariant functor, except that it "turns morphisms around" ("reverses all the arrows"). More specifically, every morphism $f : X \to Y$ in $\mathbf{C}$ must be assigned to a morphism $F^1(f) : F^0(Y) \to F^0(X)$ in $\mathbf{D}$. In other words, a contravariant functor is a covariant functor from the opposite category $\mathbf{C}^{\mathrm{op}}$ to $\mathbf{D}$. Opposite category $\mathbf{C}^{\mathrm{op}}$ has the same objects as the category $\mathbf{C}$. Given a composition of arrows $f \circ g$ in $\mathbf{C}$ its opposite arrow in $\mathbf{C}^{\mathrm{op}}$ is $(f \circ g)^{\mathrm{OP}} = g^{\mathrm{OP}} \circ f^{\mathrm{OP}}$.

Every functor preserves epi, mono and isomorphisms.

A locally small category $\mathbf{C}$, for any object X, has a *representable functor* $F_X = \mathbf{C}(X, _) : \mathbf{C} \to \mathbf{Set}$, such that for any object Y, $F_X(Y) = \mathbf{C}(X, Y)$, and any arrow $f : Y \to Y'$ gives by composition a function $F_X(f) : \mathbf{C}(X, Y) \to \mathbf{C}(X, Y')$.

A functor $F : \mathbf{C} \to \mathbf{D}$ is called *full* if for every two objects A and B of $\mathbf{C}$, $F : \mathbf{C}(A, B) \to \mathbf{D}(F(A), F(B))$ is a surjection. A functor F is called *faithful* if this map is always injective. A functor F is called *endofunctor* if $\mathbf{D} = \mathbf{C}$.

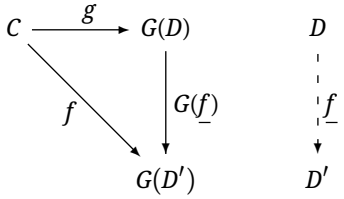
A functor F *reflects* a property P if whenever the F-image of something (object, arrow, etc.) has P, then that that something has P. For example, the faithful functor reflects epimorphisms and monomorphisms. Each endofunctor $F : \mathbf{C} \to \mathbf{C}$ defines the algebras and coalgebras (the left and right commutative diagrams):



(A.4)

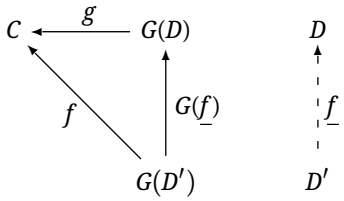
so that the morphism $f : (A, h) \to (B, k)$, which represents the commutative diagram on the left, is a morphism between F-algebras (A, h) and (B, k). For example, let F : **Set** $\to$ **Set** be the polynomial endofunctor, such that for the set A of all reals, $F(A) = A+A+A^2+A^2$ (where + is a disjoint union) we can represent the algebra for operators $\{\exp : A \to A, \log : A \to A, \max : A^2 \to A, \min : A^2 \to A\}$, where A^2 is the Cartesian product $A \times A$, by the morphism $h = [\exp, \log, \max, \min] : (A + A + A^2 + A^2) \to A$ in **Set**, i. e., by the F-algebra (A, h). Analogously, the morphism $f_1 : (A, h_1) \to (B, k_1)$, which represents the commutative diagram on the right, is a morphism between F-coalgebras (A, h_1) and (B, k_1).

Let $G : \mathbf{D} \to \mathbf{C}$ be a functor, and C an object of **C**. A *universal arrow from C to G* is a pair (D, g) where D is an object of **D** and $g : C \to G(D)$ an morphism in **C** such that, for any object D' of **D** and morphism $f : C \to G(D')$, there exists a unique morphism $\underline{f} : D \to D'$ in **D** such that $f = G(\underline{f}) \circ g$. Diagrammatically,



(A.5)

The dual concept of the universal arrow (called *couniversal arrow* as well) from functor G to object C (opposite to previous) is a pair (D, g) with opposite direction of arrows



(A.6)

As an example for the couniversal arrows, let us consider for the functor $G : \mathbf{D} \to \mathbf{C}$ to be the standard diagonal functor when $\mathbf{C} = \mathbf{D} \times \mathbf{D}$, that is equal to

$$\triangle : \mathbf{D} \to \mathbf{D} \times \mathbf{D},$$

such that for each arrow $h : A \to B$ in **D**, we have that

$$\triangle(A) \triangleq (A, A) \quad \text{and} \quad \triangle(h) \triangleq (h, h). \tag{A.7}$$

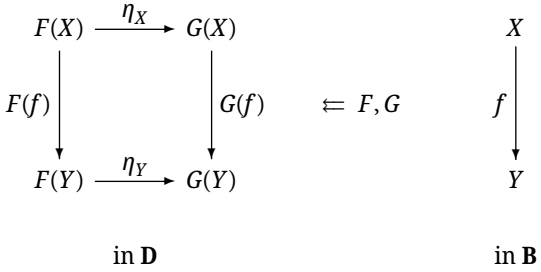
So, in the scheme (A.5) the pair (D, g) is a couniversal arrow from $\triangle$ to C.

A *natural transformation* is a relation between two functors. Functors often describe "natural constructions" and natural transformations then describe "natural homomorphisms" between two such constructions. Sometimes two quite different con-

structions yield "the same" result; this is expressed by a natural isomorphism between the two functors.

If F and G are (covariant) functors between the categories **B** and **D**, then a natural transformation η from F to G associates to every object X in **B** a morphism $\eta_X : F(X) \to G(X)$ in **D** such that for every morphism $f : X \to Y$ in **B**, we have $\eta_Y \circ F(f) = G(f) \circ \eta_X$.

This means that the following diagram is commutative:



(A.8)

and from this natural transformation we can obtain also the following two:

1. For each functor $L : \mathbf{A} \dashrightarrow \mathbf{B}$, the natural transformation $\eta L : F \circ L \dashrightarrow G \circ L$ is defined, for each object X in **A** by

$$(\eta L)_X \triangleq \eta_{LX} = (\eta L^0)(X). \tag{A.9}$$

2. For each functor $L : \mathbf{D} \dashrightarrow \mathbf{C}$, the natural transformation $L\eta : L \circ F \dashrightarrow L \circ G$ is defined, for each object Y in **B** by

$$(L\eta)_Y \triangleq L\eta_Y = (L^1\eta)(Y). \tag{A.10}$$

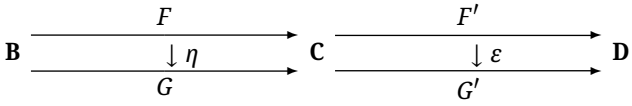
Example. Let P and Q be two preorders, regarded as categories. A functor F from P to Q is a monotone function, and there exists a unique natural transformation between two such, $\eta : F \dashrightarrow G$, exactly if $F(X) \leq G(X)$ for all X in P. □

If $\eta : F \dashrightarrow G$ and $\varepsilon : G \dashrightarrow H$ are natural transformations between functors $F, G, H : \mathbf{C} \to \mathbf{D}$, then we can compose them to get a natural transformation $\varepsilon \bullet \eta : F \dashrightarrow H$.

This is done componentwise: $(\varepsilon \bullet \eta)_X = \varepsilon_X \circ \eta_X : F(X) \to H(X)$.

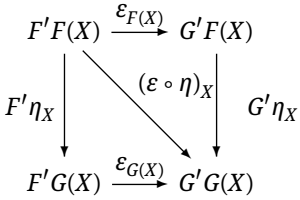
This *vertical composition* of natural transformation is associative and has an identity, and allows one to consider the collection of all functors $\mathbf{C} \to \mathbf{D}$ itself as a category $\mathbf{D}^{\mathbf{C}}$ of all functors from **C** to **D**.

There is also a *horizontal composition* (or star product) of natural transformations, given the functors $F, G : \mathbf{B} \to \mathbf{C}$, $F', G' : \mathbf{C} \to \mathbf{D}$, and two natural transformations $\eta : F \dashrightarrow G$ and $\varepsilon : F' \dashrightarrow G'$, shown by following scheme:



(A.11)

defined by $\varepsilon \circ \eta : F' \circ F \stackrel{\cdot}{\longrightarrow} G' \circ G$, between the composed functors $F' \circ F, G' \circ G : \mathbf{B} \to \mathbf{D}$. It holds that $\varepsilon F = \varepsilon \circ \mathrm{id}_F$ and $F'\eta = \mathrm{id}_{F'} \circ \eta$. The arrow components of horizontal composition $\varepsilon \circ \eta$, for any object X of $\mathbf{B}$ is the diagonal arrow of the following commutative diagram:



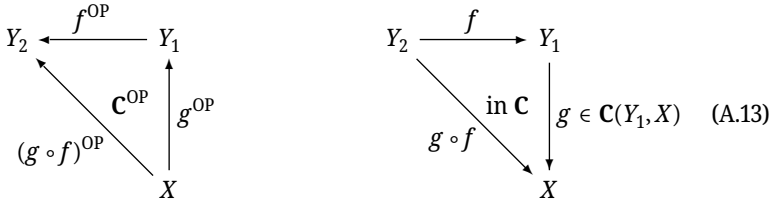
i. e.,

$$(\varepsilon \circ \eta)_X = G'\eta_X \circ \varepsilon_{F(X)} = \varepsilon_{G(X)} \circ F'\eta_X \tag{A.12}$$

The two functors F and G are called naturally isomorphic if there exists a natural transformation η from F to G such that η_X is an isomorphism for every object X in $\mathbf{C}$.

The Yoneda lemma is one of the most famous basic results of category theory. It describes representable functors in functor categories. The Yoneda embedding for any given category $\mathbf{C}$ is given by the covariant functor $H : \mathbf{C} \to \mathbf{Set}^{\mathbf{C}^{\mathrm{OP}}}$ such that, for any object X in $\mathbf{C}$, we obtain the contravariant hom-functor $h_X = \mathbf{C}^{\mathrm{OP}}(X, _) : \mathbf{C}^{\mathrm{OP}} \to \mathbf{Set}$.

Consequently, for any object Y' in $\mathbf{C}^{\mathrm{OP}}$, $h_X(Y_1) = \mathbf{C}^{\mathrm{OP}}(X, Y_1)$, while for any arrow $f^{\mathrm{OP}} : Y_1 \to Y_2$ in $\mathbf{C}^{\mathrm{OP}}$, we obtain the arrow (function) in $\mathbf{Set}$, $h_X(f^{\mathrm{OP}}) : \mathbf{C}^{\mathrm{OP}}(X, Y_1) \to \mathbf{C}^{\mathrm{OP}}(X, Y_2)$ is a composition with f^{OP}, i. e., for any $g^{\mathrm{OP}} : X \to Y_1$ in $\mathbf{C}^{\mathrm{OP}}(X, Y_1)$ we obtain the arrow $h_X(f^{\mathrm{OP}})(g^{\mathrm{OP}}) = f^{\mathrm{OP}} \circ g^{\mathrm{OP}} = (g \circ f)^{\mathrm{OP}} : X \to Y_2$, such that the following diagram commutes in $\mathbf{C}^{\mathrm{OP}}$ (or, dually in $\mathbf{C}$):

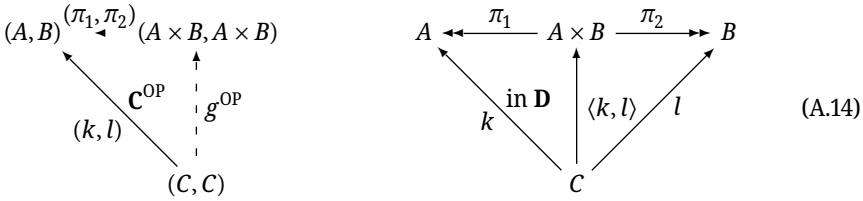


(A.13)

In the case when $f^{\mathrm{OP}} : Y_1 \to Y_2$ in $\mathbf{C}^{\mathrm{OP}}$ is a *universal arrow*, the function $h_X(f^{\mathrm{OP}}) : \mathbf{C}^{\mathrm{OP}}(X, Y_1) \to \mathbf{C}^{\mathrm{OP}}(X, Y_2)$ (an arrow in **Set**) is a bijection.

Let us consider two significant cases of Yoneda embedding for categorial *products* and *coproducts*:

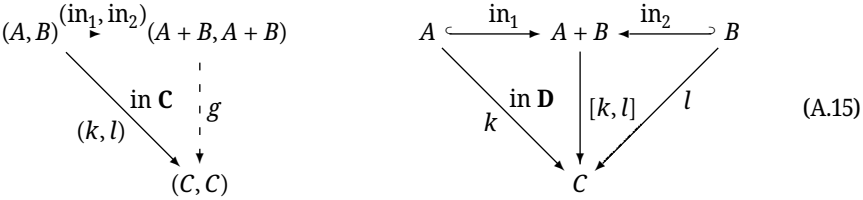
1. The universal arrows for the *products* $f^{\mathrm{OP}} = (\pi_1, \pi_2) : (A \times B, A \times B) \twoheadrightarrow (A, B)$ in $\mathbf{C}^{\mathrm{OP}} = \mathbf{D} \times \mathbf{D}$, and a given object $X = (C, C) \in \mathbf{C}^{\mathrm{OP}}$, where $\pi_1 : A \times B \twoheadrightarrow A$, $\pi_2 : A \times B \twoheadrightarrow B$ are two epimorphisms in $\mathbf{D}$ corresponding to the first and second projections, respectively. Thus, we obtain the following case for the commutative diagram above in $\mathbf{C}^{\mathrm{OP}} = \mathbf{D} \times \mathbf{D}$ (and its corresponding commutative diagram in $\mathbf{D}$ on the right), where $Y_2 = (A, B)$ and $Y_1 = (A \times B, A \times B)$, $g^{\mathrm{OP}} = (\langle k, l\rangle, \langle k, l\rangle) : (C, C) \to (A \times B, A \times B)$, with $k : C \to A$, $l : C \to B$ two arrows in $\mathbf{D}$:



(A.14)

The bijective function $h_X(f^{\mathrm{OP}}) : \mathbf{C}^{\mathrm{OP}}(X, Y_1) \to \mathbf{C}^{\mathrm{OP}}(X, Y_2)$ in this case means that for any arrow $(k, l) : (C, C) \to (A, B)$ in $\mathbf{C}^{\mathrm{OP}}(X, Y_2)$ (i. e., the couple of arrows $k : C \to A$, $l : C \to B$ in $\mathbf{D}$) there is the unique arrow $(\langle k, l\rangle, \langle k, l\rangle) : (C, C) \to (A \times B, A \times B)$ in $\mathbf{C}^{\mathrm{OP}}(X, Y_1)$ (i. e., the unique arrow $\langle k, l\rangle : C \to A \times B$ in $\mathbf{D}$).[2]

2. Universal arrows for the *coproducts* $f = (\mathrm{in}_1, \mathrm{in}_2) : (A, B) \hookrightarrow (A + B, A + B)$ in $\mathbf{C} = \mathbf{D} \times \mathbf{D}$, and a given object $X = (C, C) \in \mathbf{C}$, where $\mathrm{in}_1 : A \hookrightarrow A + B$, $\mathrm{in}_2 : B \hookrightarrow A + B$ are two monomorphisms in $\mathbf{D}$ corresponding to the first and second injections, respectively. Thus, we obtain the following case for the commutative diagram above in $\mathbf{C} = \mathbf{D} \times \mathbf{D}$ (and its corresponding commutative diagram in $\mathbf{D}$ on the right), where $Y_2 = (A, B)$ and $Y_1 = (A + B, A + B)$, $g = ([k, l], [k, l]) : (A + B, A + B) \to (C, C)$, with $k : A \to C$, $l : B \to C$ two arrows in $\mathbf{D}$:



(A.15)

The bijective function $h_X(f^{\mathrm{OP}}) : \mathbf{C}(Y_1, X) \to \mathbf{C}(Y_2, X)$ in this case means that for any arrow $(k, l) : (A, B) \to (C, C)$ in $\mathbf{C}(Y_2, X)$ (i. e., the couple of arrows $k : A \to C$, $l : B \to C$ in $\mathbf{D}$) there is the unique arrow $([k, l], [k, l]) : (A + B, A + B) \to (C, C)$ in $\mathbf{C}(Y_1, X)$ (i. e., the unique arrow $[k, l] : A + B \to C$ in $\mathbf{D}$).

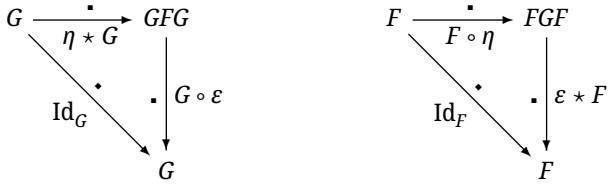
---

2 If $\mathbf{D}$ is the category **Set**, then the standard Cartesian product of A, B is a limit and we would have $\langle k, l\rangle(x) = \langle k(x), l(x)\rangle$, for each $x \in C$, explaining why the ordered pair notation $\langle k, l\rangle$ is commonly abused for the purpose of denoting that morphism.

For example, in **Set** the object $A \times B$ is the Cartesian product of the set A and set B, while $A + B$ is the disjoint union of the set A and set B.

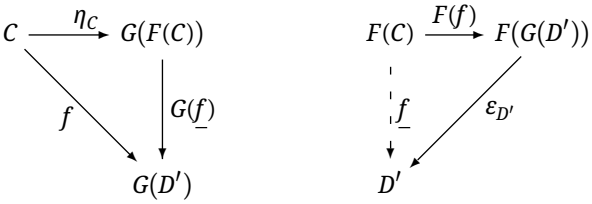
Products and coproducts are the particular cases of adjunctions. Given two functors $F : \mathbf{C} \to \mathbf{D}$, $G : \mathbf{D} \to \mathbf{C}$, we say that F is *left adjoint* to G, or G is *right adjoint* to F, if there are the natural isomorphism: $\theta : \mathbf{C}(_, G(_)) \xrightarrow{\cdot} \mathbf{D}(F(_), _)$, where $\mathbf{C}(_, G(_)) : \mathbf{C}^{\mathrm{OP}} \times \mathbf{D} \to \mathbf{Set}$ is the result of composing the bivariant hom-functor $\mathbf{C}(_, _)$ with $\mathrm{Id}_{C^{\mathrm{OP}}} \times G$, and $\mathbf{D}(F(_), _)$ is similar.

Equivalently, they are adjoint if there are two natural transformations, *counit* $\varepsilon : FG \xrightarrow{\cdot} \mathrm{Id}_D$ and *unit* $\eta : \mathrm{Id}_C \xrightarrow{\cdot} GF$, such that the following diagrams of natural transformations commute:



where $\eta \star G$ denotes the natural transformation with components (functions) $\eta_{G(X)} : G(X) \to GFG(X)$ and $G \circ \varepsilon$ denotes the natural transformation with components $G(\varepsilon_X) : GFG(X) \to G(X)$, for each object X in **D**. Thus, we have the following.

Definition 58 (Adjunction). This adjunction is denoted by tuple $(F, G, \varepsilon, \eta)$, with $\eta_C : C \to G(F(C))$ a universal arrow for each object C. In fact, for any object D' in **D** and morphism $f : C \to G(D')$ there exists a unique morphism $\underline{f} : D \to D'$, where $D = F(C)$, such that the following two adjoint diagrams commute:



(A.16)

so that η_C is a universal arrow from C into G and, dually, $\varepsilon_{D'}$ is a couniversal arrow from F into D'.

Consequently, in a given adjunction, the unit η generates a universal arrow for each object in **C** and counit ε generates a couniversal arrow for each object in **D**. In the case when **C** and **D** are poset categories, f and $\widehat{f}$ define this adjunction as a Galois connection, denoted by $F \dashv G$.

In category theory, the abstract notion of a *limit* captures the essential properties of universal constructions such as products, pullbacks and inverse limits. The dual notion of a *colimit* generalizes constructions such as disjoint unions, direct sums, coproducts, pushouts and direct limits. As for a computer science, limits have been brought to the

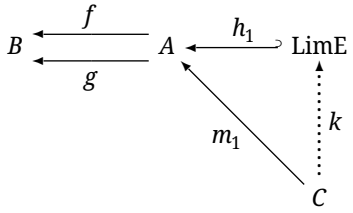
limelight mainly by the semantic investigation of recursive definition of data types: one of the early application of category theory to computer science was the solution of the recursive domain equations. This kind of equation is typical of every programming language that allows an explicit or implicit form of self-application of its data types (such as a recursive procedure call, for example).

The concept of limits embodies the general idea of universal constructions, i. e., of an entity, which has a privileged behavior among a class of objects that satisfy a certain property. The only way to define a property in the categorial language is by specifying the existence and equality of certain arrows, i. e., essentially by imposing the existence of particular *commutative diagrams* among objects inside the category.

Thus, the *diagrams* have a primary role for definition of limits. Limits and colimits in a category **C** are defined by means of diagrams in **C**. Formally, a diagram in a category **C** should be defined as a graph homomorphism from an index graph D_I to the (graph D underlying the) category **C**. If we define the sketch category $\mathbf{J} = \mathbf{Sch}(D_I)$ (derived from the index graph D_I) with the indexed objects $a_i \in \mathrm{Ob}_{\mathbf{J}}$ and morphisms $l_{jm} : a_j \to a_m$, then a commutative homomorphic diagram in **C** is represented by a functor $D = (D^0, D^1) : \mathbf{J} \to \mathbf{C}$, that is by objects in functor category $\mathbf{C}^{\mathbf{J}}$, so that the objects and arrows of the homomorphic diagram in **C** are derived by $D^0(a_j)$ and $D^1(l_{jm})$, relatively.

Limits and colimits, like the strongly related notions of universal properties and adjoint functors, exist at a high level of abstraction. From this more abstract point of view, the limits and colimits are important cases of universal arrows, where the involved functor is "diagonal" (constant), as is in the details shown in Sections 2.3 and 2.4.

For example, the *equalizers* are a particular case of limits (of pullbacks). For each pair of simple morphisms $f, g : A \to B$, the LimE is a limit object (up to isomorphism) of the diagram $B \overset{f}{\underset{g}{\leftarrow}} A$, so that the following diagram is an equalizer:



(A.17)

with $f \circ h_1 = g \circ h_1$ where h_1 is a monomorphism (every equalizer is monic), and for any arrow m_1 such that $f \circ m_1 = g \circ m_1$, there is a unique arrow k such that $m = h_1 \circ k$.

Each limit object is a vertex of a *limit cone* of arrows toward the diagram for which this object is the limit. In the example of equalizer above, object LimE is the vertex of the cone toward the diagram $B \overset{f}{\underset{g}{\leftarrow}} A$, composed by the arrows toward each object of diagram, $h_1 : \mathrm{LimE} \to\!\!\!\!\to A$ and $h_2 : \mathrm{LimE} \to\!\!\!\!\to B$ such that $h_2 = f \circ h_1 = g \circ h_1$. So, generally, if the diagram is represented by a functor $D(D^0, D^1) : \mathbf{J} \to \mathbf{C}$, then the cone

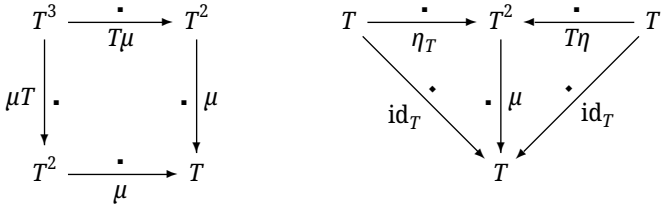
is composed by the arrows $h_i : \text{Lim} \to D^0(a_i)$ from limit object Lim into the diagram's objects $D^0(a_i)$ for each indexed object a_i in small index category **J**. In this case of the equalizer, we have that $A = D^0(a_1)$, $B = D^0(a_2)$ and $f = D^1(l_{12}), f = D^1(l'_{12})$, relatively.

In the case of limits, for another cones from an object C into the same diagram, we have a unique arrow $k : C \to \text{Lim}$, and this property defines the concept of limits. In the case above in commutative diagram (A.17), we have the cone from an object C into the diagram, composed by the arrows from C into objects of the diagram, $m_1 : C \to A$ and $m_2 = f \circ m_1 = g \circ m_1 : C \to B$. So, we have the unique arrow between the vertex C of this cone into the vertex (limit) of the limit cone, $k : C \to \text{LimE}$.

For the dual case of colimits, the arrows of cones are inverted, and are called cocones, and also the unique arrow between cocones is oriented from the colimit cocone into each other cocone over the same diagram.

Two categories **C** and **D** are *equivalent* if they are adjoint with the unit and counit whose components are all isomorphisms.

Given an adjunction $(F, G, \varepsilon, \eta)$, let us look at the endofunctor $T = GF : \mathbf{C} \to \mathbf{C}$. We have a natural transformation $\eta : \text{id}_{\mathbf{C}} \xrightarrow{\cdot} T$ and a natural transformation $\mu : T^2 \xrightarrow{\cdot} T$ with components for each object C, $\mu_C = G(\varepsilon_{F(C)}) : T^2C \to TC$ (Here, T^2 denotes the composition TT and T^3 the composition TTT). Furthermore, the equalities (commutative diagrams of natural transformations)



(A.18)

hold. Here, $(T\mu)_C = T(\mu_C) : T^3C \to TC$ and $(\mu T)_C = \mu_{TC} : T^3C \to TC$ (similarly for ηT and $T\eta$).

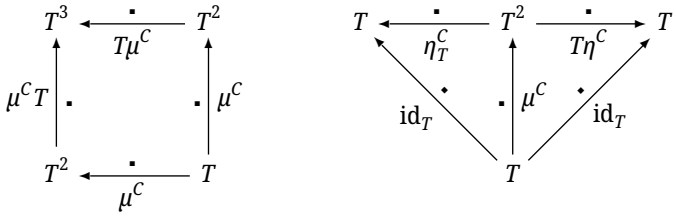
A triple (T, η, μ) satisfying these equalities is called a *monad*.

The notion of a monad is one of the most general mathematical notions. For instance, *every* algebraic theory, i. e., every set of operations satisfying equational laws, can be seen as a monad (which is also a *monoid* in a category of endofunctors of a given category: the "operation" μ being the associative multiplication of this monoid and η its unit).

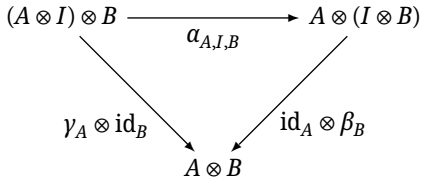
Thus, monoid laws of the monad do subsume all possible algebraic laws.

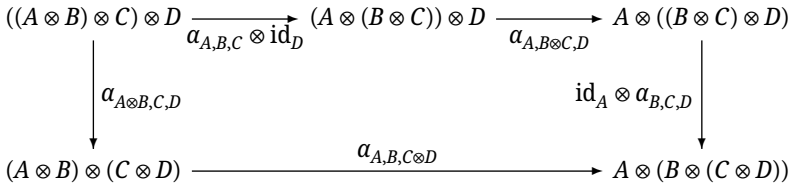
We will use monads [12, 26, 93] for giving *denotational semantics to database mappings*, and more specifically as a way of modeling computational/collection types [94–97].

A dual structure to a monad is a comonad (T, η^C, μ^C) such that all arrows in the commutative monad's diagrams above are inverted, so that we obtain:



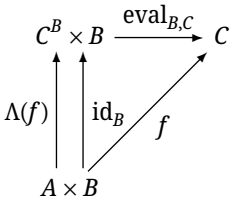
A *monoidal category* is a structure $(\mathbf{C}, \otimes, I, \alpha, \beta, \gamma)$ where $\mathbf{C}$ is a category, $\otimes : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$ is a functor (*tensor*), I is a distinguished object of $\mathbf{C}$ (unit), α, β, γ are natural transformations (*structural isos* for associativity, left and right identity) with components: $\alpha_{A,B,C} : (A \otimes B) \otimes C \simeq A \otimes (B \otimes C)$, $\beta_A : I \otimes A \simeq A$, $\gamma_A : A \otimes I \simeq A$, such that the following diagrams commute (coherence conditions):





A *strict* monoidal category is one for which all these natural transformations are identities.

If the category is finitely complete and has the *exponentiation* as well, then it is a Cartesian Closed Category (CCC). We say that a category has exponentiation if it has a product for any two objects, and if for any given objects B and C there is an object C^B and an arrow $\mathrm{eval}_{B,C} : C^B \times B \to C$, called an *evolution* arrow, such that for any object A and arrow $f : A \times B \to C$, there is a unique arrow $\Lambda(f) : A \to C^B$ making the following diagram commute:



(A.19)

such that $\text{eval}_{B,C} \circ (\Lambda(f) \times \text{id}_B) = f$. The assignment of the curried[3] function $\Lambda(f)$ to f establishes a bijection Λ from the arrows from $A \times B$ to C and the arrows from A to C^B. The most known CCC is the category **Set**, and the meaning of the diagram above is simple:

For each pair of elements $a \in A$ and $b \in B$, we obtain a function $h \triangleq \Lambda(f)(a) : B \to C$, such that by applying the function $\text{eval}_{B,C}$ we obtain that it applies the function h to the element b, so that $\text{eval}_{B,C}(h, b) \triangleq h(b) = f(a, b)$.

This very short introduction to category theory is dedicated to people, which did not work previously with the concepts of the category theory.

A.3 Introduction to RDB, database mappings and DB category

Database schema mappings are so important in information integration that many mapping formalisms have been proposed for different tasks. A schema mapping is a high-level *declarative specification* of the relationship between two schemas; it specifies how data structured under one schema, called source schema, is to be converted into data structured under possibly different schema, called the target schema. It the last decade, schema mappings have been fundamental components for both data exchange and data integration. In this work, we will consider the declarative schema mappings between relational databases. A widely used formalism for specifying relational-to-relational schema mappings is that of tuple generating dependencies (tgds). In the terminology of data integration, tgds are equivalent to global-and-local-as-view (GLAV) assertions. Using a language that is based on tgds for specifying (or "programming") database schema mappings has several advantages over lower-level languages, such as XSLT scripts of Java programs in that it is declarative and it has been widely used in the formal study of the semantics of data exchange and data integration. Declarative schema mapping formalisms have been used to provide formal semantics for data exchange [99], data integration [92], peer data management [121, 122], pay-as-you-go integration systems [123] and model management operators [124].

Most of the work in the data integration/exchange and peer-to-peer (P2P) framework is based on a logical point of view (particularly for the integrity constraints, in order to define the right models for certain answers) in a "local" mode (source-to-target database) where proper attention to the general "global" problem of the *compositions* of complex partial mappings, which possibly involve a high number of databases has not

3 Currying provides a way for working with functions that take multiple arguments, and using them in frameworks where functions might take only one argument. This property is inherited from lambda calculus, where multiargument functions are usually represented in curried form. A curried function is a function that takes multiple parameters one at a time, by taking the first argument, and returning a series of functions which each take the next argument until all the parameters have been fixed, and the function application can complete, at which point, the resulting value is returned.

been given. Today, this "global" approach cannot be avoided because of the necessity of P2P open-ended networks of heterogenous databases. Recently [5], we provided a *definition of a **DB** category for the database mappings,* which has to be more suitable than a generic **Set** domain category since the databases are more complex structures w. r. t. the sets and the mappings between them are so complex that they cannot be represented by a *single* function (which is one arrow in **Set**). In such an enriched categorical semantic domain, it is possible:

- To give a proper solution for a general problem of complex database-mappings and for the high-level algebra operators of the databases (merging, matching, etc.), by preserving the traditional common practice logical language for schema database mapping definitions.
- The schema mappings are not the integral parts of the standard relational-DB theory (used to define a database schema with its integrity constraints); they are the *programs* and we need an enriched denotational semantics context that is able to formally express these programs (derived by the mappings between the databases).
- Let us consider, e. g., the P2P systems or the mappings in a complex data warehouse. We would like to have a synthetic graphical representations of the database mappings and queries and to be able to develop a graphical tool for the meta-mapping descriptions of complex (and partial) mappings in various contexts, with a formal mathematical background.

Hence, we need a framework of interdatabase mappings. The main difference between the previous categorial approaches to databases and that presented in [5] is the *level* of abstraction used for the prime objects of the theory.

Another difference is methodological. In fact, the logics for relational databases are based on different kinds of First-Order Logic (FOL) sublanguages as, e. g., Description Logic, Relational Database logic, DATALOG, etc. So, the previous work on categorical semantics for the DB theory strictly follows an earlier research for categorial many-sorted FOL on the predicates with types where each attribute of a predicate has a particular sort with a given set of values. Thus, the *fibered semantics* for predicates is assumed for such a typed logic, where other basic operations as negation, conjunction and FOL quantifiers (that are algebraically connected with the Galois connection of their types, traduced by left and right adjunction of their functors in categorical translation) are defined *algebraically* in such a fibrational formulation. This algebraic method, of translation the FOL in a categorical language, is successively and directly applied to the DB theory seen as a sublanguage of the FOL. So, there are no the new results from the previous developed for the FOL in this translation of DB-theory into a categorical framework. No new particular *base category* is defined for databases (different from **Set**), as it happened in the cases, e. g., of the Cartesian Closed Categories (CCC) for typed λ-calculus, Bicartesian Closed Poset Categories for Heyting algebras, or the elementary topos (with the subobject classifier diagrams) for the intuitionistic logic [19, 125]. Basically, all previously works use the **Set** category as the base denotational semantics category, without

considering the question if such a topos is also *necessary* requirement for the database-mapping theory.

In our case, we are working with Relational Databases (RDB), and consequently with Structured Query Language (SQL), which is an extension of Codd's "Select-Project-Join + Union" (SPJRU) relational algebra [126, 127]. We assume *a view* of a database A as an observation on this database, presented as a relation (a set of tuples) obtained by a query $q(\mathbf{x})$ (SPJRU term with a list of free variables in $\mathbf{x}$), where $\mathbf{x}$ is a list of attributes of this view. Let $\mathcal{L}_A$ be the set of all such queries over A and $\mathcal{L}_A/_{\approx}$ be the quotient term algebra obtained by introducing the equivalence relation $\approx$, such that $q(\mathbf{x}) \approx q'(\mathbf{x})$ if both queries return with the same relation (view). Thus, a view can be equivalently considered as a *term* of this quotient-term algebra $\mathcal{L}_A/_{\approx}$ with carrier set of relations in A and a finite arity of their SPRJU operators whose computation returns a set of tuples of this view. If this query is a finite term of this algebra, then it is called a "finitary view" (a finitary view can have an infinite number of tuples as well).

In this coalgebraic methodological approach to databases, we consider a database instance A of a given database schema $\mathcal{A}$ (i. e., the set of relations that satisfy all integrity constraints of a given database schema) as a black box and any view (the response to a given query) is considered as an *observation*. Thus, in this framework we do not consider a categorical semantic for the free syntax algebra of a given query language, but only the resulting observations and the query-answering system of this database (an Abstract Object Type (AOT)), i. e., the *coalgebra*. Consequently, all algebraic aspects of the query language are encapsulated in the single power-view operator T, such that for a given database instance A (first object in our base database category) the object TA is the set of all possible views of this database A that can be obtained from a given query language $\mathcal{L}_A/_{\approx}$.

A functorial translation of database schema intermappings (a small graph category) into the database category **DB** is fundamentally based on a functor that represents a given *model* of this database schema intermappings theory. This functor maps a data schema of a given database into an single object of the **DB** category, i. e., a database instance A of this database schema $\mathcal{A}$ (a *model* of this database schema, composed of a set of relations that satisfy the schema's integrity constraints). The morphisms in the **DB** category are not simple functions as in the **Set** category. Thus, the category **DB** is not an elementary (standard) topos. In fact, it was shown in [128] that if we want to progress to more expressive sketches w. r. t. the original Ehresmann's sketches for diagrams with limits and coproducts, by eliminating nondatabase objects as, for example, Cartesian products of attributes or powerset objects, we need *more expressive arrows* for sketch categories (diagram predicates in [128] that are analog to the approach of Makkai in [129]). As we progress to a more abstract vision in which objects are the whole databases, following the approach of Makkai, in [5] we obtained more complex arrows in this new basic **DB** category for databases in which objects are just the database instances (each object is a set of relations that compose this database instance). Such arrows are not just simple functions as in the case of the **Set** category but complex trees (i. e., operads) of

view-based mappings: each arrow is equivalent to the *sets of functions*. In this way, while Ehresmann's approach prefers to deal with a few fixed diagram properties (commutativity, (co)limitness), we enjoy the possibility of setting a full relational-algebra signature of diagram properties.

Different properties of this **DB** category were presented in a number of previously published papers, in initial versions, [64, 67, 112, 130, 131] as well, and it has been demonstrated that this category is a *weak monoidal topos*. The fundamental power view-operator T has been defined in [61]. For the database applications, the powerview-monad $T : \mathbf{DB} \to \mathbf{DB}$ is used to provide *denotational semantics to database mappings* [5, 67]: in order to interpret a database mappings (morphisms) in the category **DB**, we distinguish the object $A \in \mathrm{Ob}_{\mathrm{DB}}$ (a database instance of type $\mathcal{A}$) from the object TA of observations (computations of type $\mathcal{A}$ without side-effects) and take as a denotation of (view) mappings the elements of TA (which are the views of type $\mathcal{A}$). In particular, we identify A with the object of values (of type $\mathcal{A}$) and we obtain the object of observations by applying the unary type-constructor T (power-view operator) to A, and hence in **DB** we have the isomorphism $A \simeq TA$ (for this monad η is a natural isomorphism, i. e., for each object A, $\eta_A : A \simeq TA$, and μ generates the identity morphisms, i. e., $\mu_A : T^2A \to TA$ is the identity morphism id_{TA} with $T^2A = TA$). In fact, each query (the union of conjunctive queries), expressed by FOL formulae, has an equivalent finite-length *algebraic term* of the SPJRU algebra, and hence the power view-operator T can be defined by the initial SPRJU algebra of ground terms. We define this fundamental *idempotent powerview* operator T, in Section A.3.2, with the domain and codomain equal to the set of all instance databases such that for any instance-database A, the object $TA = T(A)$ denotes a database composed of the set of *all views* of A. For every object A, $A \subseteq TA$ and $TA = T(TA)$, i. e., each (element) view of database instance TA is also an element (view) of a database instance A.

The Kleisli category and the semantics of morphisms in the **DB** category, based on the monad (endofunctor T) have been presented in [66]. The semantics for merging and matching database operators based on complete database lattice, as in [132], were defined as well and presented in a number of papers cited above.

A.3.1 Basic database concepts

The database mappings, for a logic language (we assume the FOL language in Definition 55) with a set of predicate letters P, are usually defined at a schema level by:

1. A *database schema* is a pair $\mathcal{A} = (S_A, \Sigma_A)$ where S_A is a countable set of relational symbols (predicates in FOL) $r \in P$ with finite arity $n = \mathrm{ar}(r) \geq 1$ ($\mathrm{ar} : P \to \mathcal{N}$), disjoint from a countable infinite set **att** of attributes (a domain of $a \in \mathbf{att}$ is a nonempty finite subset $\mathrm{dom}(a)$ of a countable set of individual symbols **dom**, with $\mathcal{D} = \mathbf{dom} \bigcup \mathrm{SK}$). For any $r \in P$, the sort of r, denoted by tuple $\mathbf{a} = \mathrm{atr}(r) = \langle \mathrm{atr}_r(1), \ldots, \mathrm{atr}_r(n) \rangle$ where all $a_i = \mathrm{atr}_r(m) \in \mathbf{att}$, $1 \leq m \leq n$, must be distinct:

if we use two equal domains for different attributes, then we denote them by $a_i(1), \ldots, a_i(k)$ (a_i equals to $a_i(0)$). Each index ("column") i, $1 \leq i \leq \mathrm{ar}(r)$, has a distinct column name $nr_r(i) \in \mathrm{SN}$ where SN is the set of names with $nr(r) = \langle nr_r(1), \ldots, nr_r(n)\rangle$. A relation $r \in P$ can be used as an atom $r(\mathbf{x})$ of FOL with variables in $\mathbf{x}$ assigned to its columns, so that Σ_A denotes a set of sentences (FOL formulae without free variables) called *integrity constraints* of the sorted FOL with sorts in **att**. We denote the empty schema by $\mathcal{A}_\emptyset = (\{r_\emptyset\}, \emptyset)$, where $r_\emptyset$ is the relation with empty set of attributes (*truth* propositional letter in FOL, Definition 55), and we denote the set of all database schemas for a given (also infinite) set P by $\mathbb{S}$.

2. An *instance database* of a nonempty schema $\mathcal{A}$ is given by $A = (\mathcal{A}, I_T) = \{R = \|r\| = I_T(r) \mid r \in S_A\}$ where I_T is a Tarski's FOL interpretation in Definition 55, which satisfies *all* integrity constraints in Σ_A and maps a relational symbol $r \in S_A$ into an n-ary relation $R = \|r\| \in A$. Thus, an instance-database A is a set of n-ary relations, managed by relational database systems (DBMSs). Let A and $A' = (\mathcal{A}, I'_T)$ be two instances of $\mathcal{A}$, then a function $h : A \to A'$ is a *homomorphism* from A into A' if for every k-ary relational symbol $r \in S_A$ and every tuple $\langle v_1, \ldots, v_k\rangle$ of this k-ary relation in A, $\langle h(v_1), \ldots, h(v_k)\rangle$ is a tuple of the same symbol r in A'. If A is an instance database and ϕ is a sentence, then we write $A \models \phi$ to mean that A satisfies ϕ. If Σ is a set of sentences, then we write $A \models \Sigma$ to mean that $A \models \phi$ for every sentence $\phi \in \Sigma$. Thus, the set of all instances of $\mathcal{A}$ is defined by $\mathrm{Inst}(\mathcal{A}) = \{A \mid A \models \Sigma_A\}$. We denote the set of all values in A by $\mathrm{val}(A) \subseteq \mathcal{D}$. Then "atomic database" $\mathfrak{J}_A = \{\{\langle v_i\rangle\} \mid v_i \in \mathrm{val}(A)\}$ is infinite iff $\mathrm{SK} \subseteq \mathrm{val}(A)$. Note that for each $a \in \mathrm{atr}(r)$, a subset $\mathrm{dom}(a) \subseteq \mathbf{dom}$ is finite, and any introduction of Skolem constants is ordered $\omega_0, \omega_1, \ldots$
3. We consider a rule-based *conjunctive query* over a database schema $\mathcal{A}$ as an expression $q(\mathbf{x}) \longleftarrow r_1(\mathbf{u}_1), \ldots, r_n(\mathbf{u}_n)$, with finite $n \geq 0$, r_i are the relational symbols (at least one) in $\mathcal{A}$ or the built-in predicates (e. g., $\leq, =$, etc.), q is a relational symbol not in $\mathcal{A}$ and $\mathbf{u}_i$ are free tuples (i. e., one may use either variables or constants). Recall that if $\mathbf{v} = (v_1, \ldots, v_m)$ then $r(\mathbf{v})$ is a shorthand for $r(v_1, \ldots, v_m)$. Finally, each variable occurring in $\mathbf{x}$ is a *distinguished* variable that must also occur at least once in $\mathbf{u}_1, \ldots, \mathbf{u}_n$. Rule-based conjunctive queries (called rules) are composed of a subexpression $r_1(\mathbf{u}_1), \ldots, r_n(\mathbf{u}_n)$ that is the *body*, and the *head* of this rule $q(\mathbf{x})$. The Yes/No conjunctive queries are the rules with an empty head. If we can find values for the variables of the rule, such that the body is logically satisfied, then we can deduce the head-fact. This concept is captured by a notion of "valuation." The deduced head-facts of a conjunctive query $q(\mathbf{x})$ defined over an instance A (for a given Tarski's interpretation I_T of schema $\mathcal{A}$) are equal to $\|q(x_1, \ldots, x_k)\|_A = \{\langle v_1, \ldots, v_k\rangle \in \mathcal{D}^k \mid A \models \exists \mathbf{y}(r_1(\mathbf{u}_1) \wedge \cdots \wedge r_n(\mathbf{u}_n))[x_i/v_i]_{1 \leq i \leq k}\} = I^*_T(\exists \mathbf{y}(r_1(\mathbf{u}_1) \wedge \cdots \wedge r_n(\mathbf{u}_n)))$, where the $\mathbf{y}$ is a set of variables, which are not in the head of query. We recall that the conjunctive queries are monotonic and satisfiable, and that a (Boolean) query is a class of instances that is closed under isomorphism [133]. Each conjunctive query corre-

sponds to a "select-project-join" term $t(\mathbf{x})$ of SPRJU algebra obtained from the formula $\exists \mathbf{y}(r_1(\mathbf{u}_1) \wedge \cdots \wedge r_n(\mathbf{u}_n))$.

4. We consider a finitary *view* as a union of a finite set S of conjunctive queries with the same head $q(\mathbf{x})$ over a schema $\mathcal{A}$, and from the equivalent algebraic point of view, it is a "select-project-join + union" (SPJRU) finite-length term $t(\mathbf{x})$, which corresponds to union of the terms of conjunctive queries in S. In what follows, we will use the same notation for a FOL formula $q(\mathbf{x})$ and its equivalent algebraic SPJRU expression $t(\mathbf{x})$. A materialized view of an instance-database A is an n-ary relation $R = \bigcup_{q(\mathbf{x})\in S} \|q(\mathbf{x})\|_A$. Notice that a finitary view can also have an infinite number of tuples. We denote the set of all finitary materialized views that can be obtained from an instance A by TA.

We consider the views as a universal property for databases: they are the possible observations of the information contained in an instance database. We assume that *a view* of a database A as the relation (set of tuples) obtained by a "Select-Project-Join + Union" (SPJRU) query $q(\mathbf{x})$ where $\mathbf{x}$ is a list of attributes of this view. We denote by $\mathcal{L}_A$ the set of all such queries over a database A and by $\mathcal{L}_A/_{\approx}$ the quotient algebra obtained by introducing the equivalence relation $\approx$, such that $q(\mathbf{x}) \approx q'(\mathbf{x})$ if both queries result in the same relation (view). Thus, a view can be equivalently considered as a *term* of this quotient algebra $\mathcal{L}_A/_{\approx}$ with carrier set of relations in A and a finite arity of their operators, whose computation returns a set of tuples of this view. We can use them in order to establish an equivalence relation between databases.

Database category **DB**, defined in [5], is at the *instance level*, i. e., any object in **DB** is an instance database.

The problem of sharing data from multiple sources has recently received significant attention, and a succession of different architectures has been proposed, beginning with federated databases [134, 135], followed by data integration systems [92, 98, 136], data exchange systems [99, 122, 137] and Peer-to-Peer (P2P) data management systems [121, 138–142]. A lot of research has been focused on the development of logic languages for semantic mapping between data sources and mediated schemas [58, 92, 132, 143–145], and algorithms that use mappings to answer queries in data sharing systems [59, 136, 146–152].

We consider that a *mapping* between two database schemas $\mathcal{A} = (S_A, \Sigma_A)$ and $\mathcal{B} = (S_B, \Sigma_B)$ is expressed by an union of "conjunctive queries with the same head." Such mappings are called "view-based mappings," defined by a set of FOL sentences $\{\forall \mathbf{x}_i(q_{Ai}(\mathbf{x}_i) \Rightarrow q_{Bi}(\mathbf{y}_i))|$ with $\mathbf{y}_i \subseteq \mathbf{x}_i, 1 \leq i \leq n\}$, where $\Rightarrow$ is the logical implication between these conjunctive queries $q_{Ai}(\mathbf{x}_i)$ and $q_{Bi}(\mathbf{x}_i)$, over the databases $\mathcal{A}$ and $\mathcal{B}$, respectively.

Schema mappings are often specified by the source-to-target tuple-generating dependencies (tgds), used to formalize a data exchange [99], and in the data integration scenarios under a name "GLAV assertions" [92, 98]. A tgd is a logical sentence (FOL formula without free variables), which says that if some tuples satisfying certain equalities

exist in the relation, then some other tuples (possibly with some unknown values) must also exist in another specified relation.

An equality-generating dependency (egd) is a logical sentence, which says that if some tuples satisfying certain equalities exist in the relation, then some values in these tuples must be equal. Functional dependencies are egds of a special form, e. g., primary-key integrity constraints. Thus, egds are only used for the specification of integrity constraints of a single database schema, which define the set of possible models of this database. They are not used for interschema database mappings.

These two classes of dependencies together comprise the *embedded implication dependencies* (EID) [153], which seem to include essentially all of the naturally occurring constraints on relational databases (we recall that the bold symbols $\mathbf{x}, \mathbf{y}, \ldots$ denote a nonempty list of variables).

Definition 59. We introduce the following two kinds of EIDs [153]:

1. A *tuple-generating dependency (tgd)*

$$\forall \mathbf{x}\big(q_A(\mathbf{x}) \Rightarrow q_B(\mathbf{x})\big),$$

where $q_A(\mathbf{x})$ is an existentially quantified formula $\exists \mathbf{y} \phi_A(\mathbf{x}, \mathbf{y})$ and $q_B(\mathbf{x})$ is an existentially quantified formula $\exists \mathbf{z} \psi_A(\mathbf{x}, \mathbf{z})$, and where the formulae $\phi_A(\mathbf{x}, \mathbf{y})$ and $\psi_A(\mathbf{x}, \mathbf{z})$ are conjunctions of atomic formulae (conjunctive queries) over the given database schemas. We assume the safety condition, i. e., that every *distinguished* variable in $\mathbf{x}$ appears in q_A.

We will consider also the class of *weakly-full* tgds for which query answering is decidable, i. e., when $q_B(\mathbf{x})$ has no existentially quantified variables, and if each $y_i \in \mathbf{y}$ appears at most once in $\phi_A(\mathbf{x}, \mathbf{y})$.

2. An *equality-generating dependency* (egd)

$$\forall \mathbf{x}\big(q_A(\mathbf{x}) \Rightarrow (\mathbf{y} \doteq \mathbf{z})\big),$$

where $q_A(\mathbf{x})$ is a conjunction of atomic formulae over a given database schema, and $\mathbf{y} = \langle y_1, \ldots, y_k \rangle, \mathbf{z} = \langle z_1, \ldots, z_k \rangle$ are among the variables in $\mathbf{x}$, and $\mathbf{y} \doteq \mathbf{z}$ is a shorthand for the formula $(y_1 \doteq z_1) \wedge \cdots \wedge (y_k \doteq z_k)$ with the built-in binary identity predicate $\doteq$ of the FOL.

Note that a tgd $\forall \mathbf{x}(\exists \mathbf{y} \phi_A(\mathbf{x}, \mathbf{y}) \Rightarrow \exists \mathbf{z} \psi_{\mathbf{A}}(\mathbf{x}, \mathbf{z}))$ is logically equivalent to the formula $\forall \mathbf{x} \forall \mathbf{y}(\phi_A(\mathbf{x}, \mathbf{y}) \Rightarrow \exists \mathbf{z} \psi_{\mathbf{A}}(\mathbf{x}, \mathbf{z}))$, i. e., to $\forall \mathbf{x}_1(\phi_A(\mathbf{x}_1) \Rightarrow \exists \mathbf{z} \psi_A(\mathbf{x}, \mathbf{z}))$ with the set of distinguished variables $\mathbf{x} \subseteq \mathbf{x}_1$. We will use for the integrity constraints Σ_A of a database schema $\mathcal{A}$ both tgds and egds, while for the interschema mappings, between a schema $\mathcal{A} = (S_A, \Sigma_A)$ and a schema $\mathcal{B} = (S_B, \Sigma_B)$, only the tgds $\forall \mathbf{x}(q_A(\mathbf{x}) \Rightarrow q_B(\mathbf{x}))$, as follows.

Definition 60. An elementary schema mapping is a triple $(\mathcal{A}, \mathcal{B}, \mathcal{M})$ where $\mathcal{A}$ and $\mathcal{B}$ are schemas with no relational symbol in common and $\mathcal{M}$ is a set of tgds $\forall \mathbf{x}(q_A(\mathbf{x}) \Rightarrow q_B(\mathbf{x}))$,

such that $q_A(\mathbf{x})$ is a conjunctive query with conjuncts equal to relational symbols in S_A or to a formula with built-in relational symbols, $\doteq, <, >, \ldots$, while $q_B(\mathbf{x})$ is a conjunctive query with relational symbols in S_B.

An instance of $\mathcal{M}$ is an instance pair (A, B) (where A is an instance of $\mathcal{A}$ and B is an instance of $\mathcal{B}$) that satisfies every tgds in $\mathcal{M}$, denoted by $(A, B) \vDash \mathcal{M}_{AB}$. We write $\mathrm{Inst}(\mathcal{M})$ to denote all instances (A, B) of $\mathcal{M}$.

Notice that the formula with built-in predicates, in the left side of implication of a tgd, can be expressed by only two logical connectives, conjunction and negation, from the fact that implication and disjunction can be reduced to equivalent formulae with these two logical connectives. Recall that in data exchange terminology, B is a solution for A under $\mathcal{M}$ if $(A, B) \in \mathrm{Inst}(\mathcal{M})$, and that an instance of $\mathcal{M}$ satisfies all FOL formulae in $\Sigma_A \bigcup \Sigma_B \bigcup \mathcal{M}$. For a given set of FOL formulas S, we denote by $\bigwedge S$ the conjunction of all formulae in the set S.

Lemma 11. *For any given Tarski's interpretation I_T (and its extension I_T^* in* (A.1)*), i. e., a model of the schemas $\mathcal{A} = (S_A, \Sigma_A)$ and $\mathcal{B} = (S_B, \Sigma_B)$ and of the set of tgds in the mapping $\mathcal{M}$, i. e., when $I_T^*(\bigwedge \Sigma_A) = I_T^*(\bigwedge \Sigma_B) = I_T^*(\bigwedge \mathcal{M}) = t$, one has* $(\{I_T(r) \mid r \in S_A\}, \{I_T(r) \mid r \in S_B\}) \in \mathrm{Inst}(\mathcal{M})$.

Proof. Due to the fact that $I_T^*(\bigwedge \Sigma_A) = t$ means that all integrity constraints of $\mathcal{A}$ are satisfied, one has that $A = \{I_T(r) \mid r \in S_A\}$ is an instance (model) of $\mathcal{A}$. The same holds for the schema $\mathcal{B}$, so that $B = \{I_T(r) \mid r \in S_B\}$ is an instance of $\mathcal{B}$.

From the fact that $I_T^*(\bigwedge \mathcal{M}) = t$, each tgd $\phi \in \mathcal{M}_{AB}$ is satisfied, i. e., $I_T^*(\phi) = t$, and hence $(A, B) \vDash \mathcal{M}$, i. e., $(A, B) \in \mathrm{Inst}(\mathcal{M})$. □

The formulae (tgds) in the set $\mathcal{M}$ express the constraints that an instance (A, B) over the schemas $\mathcal{A}$ and $\mathcal{B}$ must satisfy. We assume that the satisfaction relation between formulae and instances is preserved under isomorphism, which means that if an instance satisfies a formula then every isomorphic instance also satisfies that formula.

This is a mild condition that is true for all standard logical formalisms, such as first-order logic, second-order logic, fixed-point logics and infinitary logics.

Thus, such formulae represent the queries in the sense of Chandra and Harel [133]. An immediate consequence of this property is that $\mathrm{Inst}(\mathcal{M})$ is closed under isomorphism.

Remark. Differently from [144, 154, 155], each formula in $\mathcal{M}$ contains the relational symbols of both source and target schema (the integrity constraints are contained in their schemas), in order to represent an interschema mapping graphically as a graph edge $\mathcal{M}_{AB} : \mathcal{A} \to \mathcal{B}$ as in standard mathematical denotation of a mapping. □

The problem of computing semantic mappings [124, 154–158], given a semantic mapping $\mathcal{M}_{AB}$ between data schemas $\mathcal{A}$ and $\mathcal{B}$, and $\mathcal{M}_{BC}$ between $\mathcal{B}$ and $\mathcal{C}$, generally was to answer if it is possible to generate a direct semantic mapping $\mathcal{M}_{AC}$ (possibly in the same logic language formalism) between $\mathcal{A}$ and $\mathcal{C}$ that is "equivalent" to the original

mappings. Here, "equivalent" means that for any query in a given class of queries Q and for any instance of data sources, using the direct mapping yields exactly the same answer that would be obtained by the two original mappings [154].

The semantics of the composition of the schema mappings proposed by Madhavan and Halevy [154] was a significant first step. However, it suffers from certain drawbacks that are caused by the fact that this semantics is given relative to a class of queries. In this setting, the set of formulae specifying a composition $\mathcal{M}_{AC}$ of $\mathcal{M}_{AB}$ and $\mathcal{M}_{BC}$ relative to a class Q of queries need not be unique up to logical equivalence, even when the class Q of queries is held fixed. It was shown [155] that this semantics is rather fragile, because a schema mapping $\mathcal{M}_{AC}$ may be a composition of $\mathcal{M}_{AB}$ and $\mathcal{M}_{BC}$ when Q is the class of conjunctive queries, but may fail to be composition of these two (inter)schema mappings when Q is the class of conjunctive queries with inequalities.

We use for the integrity constraints Σ_A of a database schema $\mathcal{A}$ both tgds and egds, while for the interschema mappings, between a schema $\mathcal{A} = (S_A, \Sigma_A)$ and a schema $\mathcal{B} = (S_B, \Sigma_B)$, only the tgds $\forall \mathbf{x}(q_A(\mathbf{x}) \Rightarrow q_B(\mathbf{x}))$. The so-called second-order tgds (SO tgds) has been introduced in [155] as follows.

Definition 61 ([155]). Let $\mathcal{A}$ be a source schema and $\mathcal{B}$ a target schema. A second-order tuple-generating dependency (SO tgd) is a formula of the form:

$$\exists \mathbf{f}((\forall \mathbf{x}_1(\phi_1 \Rightarrow \psi_1)) \wedge \cdots \wedge (\forall \mathbf{x}_n(\phi_n \Rightarrow \psi_n))),$$

where

1. Each member of the tuple $\mathbf{f}$ is a functional symbol. Each variable in $\mathbf{x}_i$ appears in some atomic formula of ϕ_i.
2. Each ϕ_i is a conjunction of:
   - atomic formulae of the form $r_A(y_1, \ldots, y_k)$, where $r_A \in S_A$ is a k-ary relational symbol of schema $\mathcal{A}$ and $y_1, \ldots, y_k$ are variables in $\mathbf{x}_i$, not necessarily distinct;
   - the formulae with conjunction and negation connectives and with built-in predicate's atoms of the form $t \odot t'$, $\odot \in \{\doteq, <, >, \ldots\}$, where t and t' are the terms based on $\mathbf{x}_i$, $\mathbf{f}$ and constants.
3. Each ψ_i is a conjunction of atomic formulae $r_B(t_1, \ldots, t_m)$ where $r_B \in S_B$ is an m-ary relational symbol of schema $\mathcal{B}$ and $t_1, \ldots, t_m$ are terms based on $\mathbf{x}_i$, $\mathbf{f}$ and constants.

Notice that each constant a in an atom on the left-hand side of implications must be substituted by new fresh variable y_i and by adding a conjunct $(y_i = a)$ in the left-hand side of this implication, so that such atoms will have only the variables (condition 2 above). For the empty set of tgds, we will use the SOtgd tautology $r_\emptyset \Rightarrow r_\emptyset$ where $r_\emptyset$ is the relation with empty set of attributes (*truth* propositional letter in FOL, Definition 55 in Section A.1.2 in the Appendix). The fourth condition is a "safety" condition, analogous to that made for (first-order) tgds. It is easy to see that every tgd is equivalent to one SOtgd without equalities. For example, let σ be the tgd,

$$\forall x_1 \ldots \forall x_m(\phi_A(x_1,\ldots,x_m) \Rightarrow \exists y_1 \ldots \exists y_n \psi_B(x_1,\ldots,x_m,y_1,\ldots,y_n)).$$

It is logically equivalent to the following SOtgd without equalities, which is obtained by Skolemizing existential quantifiers in σ:

$$\begin{aligned}\exists f_1 \ldots \exists f_n(\forall x_1 \ldots \forall x_m(\phi_A(x_1,\ldots,x_m)\\ \Rightarrow \psi_B(x_1,\ldots,x_m,f_1(x_1,\ldots,x_m),\ldots,f_n(x_1,\ldots,x_m)))).\end{aligned}$$

Given a finite set S of tgds of an interschema mapping $\mathcal{M}$, we can find a *single* SOtgd that is equivalent to S by taking, for each tgd σ in S, a conjunct of the SOtgd to capture σ as described above (we use disjoint sets of function symbols in each conjunct, as before).

The integrity constraints of a schema $\mathcal{A}$ are transformed into schema mappings from $\mathcal{A}$ into the particular "FOL-identity" schema $\mathcal{A}_\top = (\{r_\top\}, \emptyset)$ with empty set of integrity constraints (introduced in [5], Section 4.1.1 for categorial semantics of database schemas) where its binary built-in relation $r_\top$ corresponds to FOL identity (so that the ground atom $r_\top(a,a)$ is true for each $a \in \mathcal{D}$ while, e. g., the ground atom $r_\top(0,1)$ *is false*), based on the two following lemmas [5].

Lemma 12. *Any normalized tgd constraint of a schema* $\mathcal{A} = (S_A, \Sigma_A)$, $\forall \mathbf{x}(\phi_A(\mathbf{x}) \Rightarrow r(\mathbf{t})) \in \Sigma_A^{\mathrm{egd}} \subseteq \Sigma_A$, *where* $\mathbf{t}$ *is a tuple of terms with variables in* $\mathbf{x}$ *and* $r \in S_A$, *is logically equivalent to the FOL sentence* $\forall \mathbf{x}((\phi_A(\mathbf{x}) \wedge \neg r(\mathbf{t})) \Rightarrow r_\top(0,1))$.

Lemma 13. *Any egd* $\forall \mathbf{x}(\phi_A(\mathbf{x}) \Rightarrow (\mathbf{y} \doteq \mathbf{z})) \in \Sigma_A^{\mathrm{egd}} \subseteq \Sigma_A$ *of a given schema database* $\mathcal{A} = (S_A, \Sigma_A)$ *where* $\mathbf{y} = \langle x_{j_1}, \ldots, x_{j_k}\rangle \subseteq \mathbf{x}$ *and* $\mathbf{z} = \langle x_{l_1}, \ldots, x_{l_k}\rangle \subseteq \mathbf{x}$ *such that* $j_i \neq l_i$ *for* $1 \leq i \leq k$, *is logically equivalent to the FOL formula:*

$$\forall \mathbf{x}((\phi_A(\mathbf{x}) \wedge (\mathbf{y} \neq \mathbf{z})) \Rightarrow r_\top(0,1)),$$

where $\mathbf{y} \neq \mathbf{z}$ *is an abbreviation for the formula* $(x_{j_1} \neq x_{l_1}) \vee \cdots \vee (x_{j_k} \neq x_{l_k})$.

Consequently, for the set of integrity constraints $\Sigma_A = \Sigma_A^{\mathrm{tgd}} \bigcup \Sigma_A^{\mathrm{egd}}$, of a RDB schema $\mathcal{A} = (S_A, \Sigma_A)$, we construct, by conjunction the SOtgd in Σ_A denoted by a formula Φ, and the schema "truth-mapping" $\top_{A A_\top} = \{\Phi\} : \mathcal{A} \to \mathcal{A}_\top$, as explained in [5], Section 4.1.1, Proposition 15. We recall that the "truth-mapping" does not transfer any tuple of data from the source to the target object.

A.3.2 Database observations: Idempotent power-view operator

We consider the views as a universal property for the databases: they are the possible observations of the information contained in an instance database and we can use them in order to establish an equivalence relation between databases.

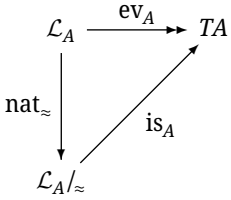
In the theory of algebraic specifications, an Abstract Data Type (ADT) is specified by a set of operations (constructors) that determine how the values of the carrier set

are built up and by a set of formulae (in the simplest case, the equations) stating which values should be identified. In the standard initial algebra semantics, the defining equations impose a congruence on the initial algebra. Dually, a coalgebraic specification of a class of systems, i. e., Abstract Object Types (AOT), is characterized by a set of operations (destructors) that specify what can be observed out of a system-state (i. e., an element of the carrier) and how a state can be transformed to a successor state.

We will introduce a class of coalgebras for database query-answering systems for a given instance-database A of a schema $\mathcal{A}$ by Definition 62. They will be presented in an algebraic style by providing a cosignature. In particular, the sorts include a single "hidden sort," corresponding to the carrier of a coalgebra, and other "visible" sorts for the inputs and outputs with a given fixed interpretation. Visible sorts will be interpreted as the sets without any algebraic structure defined on them. For us, the coalgebraic terms, built by operations (destructors), are interpreted by the basic *observations,* which one can make on the states of a coalgebra.

Input sorts for a given instance-database A is a countable set $\mathcal{L}_A$ of the union of a finite set S of conjunctive *finite-length* queries $q(\mathbf{x})$ (with the same head with a finite tuple of variables $\mathbf{x}$) so that $R = \mathrm{ev}_A(q(\mathbf{x})) = \bigcup_{q(\mathbf{x})\in S} \|q(\mathbf{x})\|_A$ is the relation (a materialized view) obtained by applying this query to A.

Each query (FOL formula introduced in Definition 55) has an equivalent finite-length *algebraic term* of the SPJRU algebra (or equivalent to it, SPCU algebra, Chapter 4.5, 5.4 in [127]) as shortly introduced in the previous section, and hence the power view-operator T can be defined by the initial SPRJU algebra of ground terms. We define this fundamental *idempotent power-view* operator T, with the domain and codomain equal to the set of all instance databases such that for any instance-database A, the object $TA = T(A)$ denotes a database composed of the set of *all views* of A. The object TA, for a given instance-database A, corresponds to the carrier of the quotient-term Lindenbaum algebra $\mathcal{L}_A/_{\approx}$, i. e., the set of the *equivalence classes* of queries (such a query is equivalent to a term in $\mathcal{T}_P X$ of an SRRJU relational algebra Σ_R, with the select, project, join and union operators, with relational symbols of a database schema $\mathcal{A}$). More precisely, TA is "generated" from A by this quotient-term algebra $\mathcal{L}_A/_{\approx}$ and a given evaluation of queries in $\mathcal{L}_A$, $\mathrm{ev}_A : \mathcal{L}_A \longrightarrow\!\!\!\!\!\rightarrow TA$, which is surjective function. From the factorization theorem, there is a unique bijection $\mathrm{is}_A : \mathcal{L}_A/_{\approx} \to TA$ such that



commutes, where the surjective function $\mathrm{nat}_{\approx} : \mathcal{L}_A \to \mathcal{L}_A/_{\approx}$ is a natural representation for the equivalence of the queries $\approx$. For every object A, $A \subseteq TA$ and $TA = T(TA)$, i. e., each

(element) view of database instance TA is also an element (view) of a database-instance A. It is easy to verify that T corresponds to the closure operator Sg on A (introduced in Section A.1) for the Σ_R relational (SPJRU) algebra, but with relations instead of variables (i. e., relational symbols). Notice that when A has a finite number of relations, but at least one relation with an infinite number of tuples, then TA has an infinite number of relations (i. e., views of A), and hence can be an infinite object.

Based on the theory of database observations and its power-view operator T, defined in Section A.3.2, the output sort of this database AOT is the set TA of all resulting views (i. e., resulting n-ary relations) obtained by computation of queries $q(\mathbf{x}) \in \mathcal{L}_A$. It is considered as the carrier of a coalgebra as well.

Definition 62. AOT for a Database query-answering system, for a given instance-database A, is a pair $(S, \Sigma_{\mathrm{AOT}})$ such that:

1. The carrier set $S = (X_A, \mathcal{L}_A, \underline{Y})$ of the sorts where X_A is a hidden sort (a set of states of this database system), $\mathcal{L}_A$ is an input sort (a set of the unions of conjunctive queries over A), and $\underline{Y}$ is the set of all finitary relations for a given universe $\mathcal{D}$.
2. The following mappings:
   - 2.1 A `method` Next : $X_A \times \mathcal{L}_A \to X_A$ that corresponds to an execution of a next query $q(\mathbf{x}) \in \mathcal{L}_A$ in a current state $s \in X_A$ of a database A, such that a database A passes to the next state; so, we introduce the destructor $\Lambda(\mathrm{Next}) : X_A \to X_A^{\mathcal{L}_A}$, where Λ is the currying operator from lambda calculus used in diagram A.19 in Section A.2.
   - 2.2 An `attribute` Out : $X_A \times \mathcal{L}_A \to \underline{Y}$ such that for each $s \in X_A$, $q(\mathbf{x}) \in \mathcal{L}_A$, $\mathrm{Out}(s, q(\mathbf{x}))$ is a relation computed by a query $q(\mathbf{x})$; analogously, we introduce the second destructor $\Lambda(\mathrm{Out}) : X_A \to \underline{Y}^{\mathcal{L}_A}$.

Then the abstract object type for a query-answering system is given by a coalgebra with the signature $\Sigma_{\mathrm{AOT}} = \{\Lambda(\mathrm{Next}), \Lambda(\mathrm{Out})\}$:

$$\langle \Lambda(\mathrm{Next}), \Lambda(\mathrm{Out}) \rangle : X_A \to FX_A$$

of the polynomial endofunctor $F \triangleq (_)^{\mathcal{L}_A} \times \underline{Y}^{\mathcal{L}_A} : \mathbf{Set} \to \mathbf{Set}$.

In an object-oriented terminology, the coalgebras are expressive enough in order to specify the parametric methods and the attributes for a database (conjunctive) query answering systems. In a transition system terminology, such coalgebras can model a deterministic, nonterminating, transition system with inputs and outputs. In [159], a complete equational calculus for such coalgebras of restricted class of polynomial functors has been defined.

Here, we will consider only the database query-answering systems without the side effects. That is, the obtained results (views) *will not be materialized* as a new relation of this database A but only visualized. Thus, when a database answers to a query, it remains in the same initial state. Thus, the set X_A is a singleton $\{A\}$ for a given database

A and, consequently, it is isomorphic to the terminal object 1 in the **Set** category. As a consequence, from $1^{\mathcal{L}_A} \simeq 1$, we obtain that a method Next is just an identity function id : $1 \to 1$. Thus, the only interesting part of this AOT is the attribute part Out : $X_A \times \mathcal{L}_A \to \underline{Y}$, with the fact that $X_A \times \mathcal{L}_A = \{A\} \times \mathcal{L}_A \simeq \mathcal{L}_A$.

Consequently, we obtain an attribute mapping Out : $\mathcal{L}_A \to \underline{Y}$, whose graph is equal to the query-evaluation surjective mapping $\text{ev}_A : \mathcal{L}_A \twoheadrightarrow TA$, introduced in the previous diagram.

Corollary 28. *A canonical method for the construction of the power-view database TA can be obtained by an abstract data-object type* (S, Σ_{AOT}) *for a query-answering system without side-effects as follows:*

$$TA \triangleq \{\text{Out}(q_i(\mathbf{x})) \mid q_i(\mathbf{x}) \in \mathcal{L}_A\}.$$

This corollary is a direct proof that the power-view database operator T, introduced previously, represents the *observational* point of view for the instance databases. This mapping $\text{ev}_A : \mathcal{L}_A \twoheadrightarrow TA$ can be used as a semantic foundation for the database mappings.

A.3.3 Logic versus algebras: Categorification by operads

In what follows, we will work with the typed operads, first developed for a purpose of homotopy theory [160–162], having a set $\mathbb{R}$ of types (each relation symbol is a type), or "R-operads" for short.

The basic idea of an R-operad O is that, given types $r_1, \ldots, r_k, r \in \mathbb{R}$, there is a set $O(r_1, \ldots, r_k, r)$ of abstract k-ary "operations" with inputs of type $r_1, \ldots, r_k$ and output of type r. In short, an operad describes a family of composable operations with multiple inputs and one output, satisfying several intuitive properties like associativity of composition and permutability of the inputs.

Categorification is the process of finding category-theoretic analog of set-theoretic concepts, by introducing "n-categories," algebraic structures having objects, morphisms between objects, 2-morphisms between morphisms and so on up to n-morphisms. In [163], it is demonstrated the existence of close relation between n-categories and the homotopy theory, based on theory of operads. As we will see in our case of base **DB** category used for denotational semantics of database mappings, this **DB** category is a n-category as well, but with a particular property that each arrow can be equivalently represented by an object (categorial symmetry); thus, higher-order arrows are represented just by ordinary 1-arrows in **DB**. Basically, the work presented in this book is the categorification of the logic theory of the database mappings.

Remark. In our case, the input types are the relational symbols of the source database of a given schema mapping (represented by a SOtgd) while the output type is a relation

symbol of the target database. Each single abstract "operation" is derived from an logic implication from a conjunctive query over the source database schema into a particular relation of the target database schema, and represents the SPJRU term of the relational algebra correspondent to the left-side (conjunctive query over source schema) of this implication, labeled by a node q in the left tree (an operation).

In what follows, we will present an algorithm *MakeOperads* that transform a given SOtgd (of a given mapping from a source to a target schema database) into the set of such abstract operations with input types in the source schema and output types.

We can visualize such an operation as a tree with only one node. In an operad, we can obtain new operations from old ones by composing them: it can be visualized in terms of trees (Figure A.1). We can obtain the new operators from old ones by permuting arguments, and there is a unary "identity" operation of each type. Finally, we insist on a few plausible axioms: the identity operations act as identities for composition, permuting arguments is compatible with composition, and *composition is associative*. Thus, formally, we have the following.

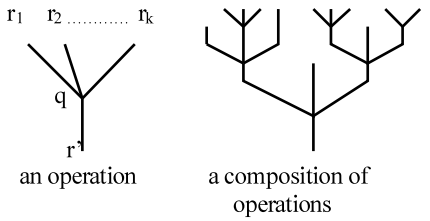


Figure A.1: Operations of an R-operad.

Definition 63. For any set $\mathbb{R}$ of relation symbols with finite arity (the predicate letters in Definition 55), an R-operad O consists of:

1. for any $r_1, \ldots, r_k, r \in \mathbb{R}$, a set $O(r_1, \ldots, r_k, r)$. For any $r \in \mathbb{R}$, an element $1_r \in O(r, r)$. We denote the empty type by $r_\emptyset$, with $\mathrm{ar}(r_\emptyset) = 0$ and $1_{r_\emptyset} \in O(r_\emptyset, r_\emptyset)$
2. for any $f \in O(r_1, \ldots, r_k, r)$ and any $g_1 \in O(r_{11}, \ldots, r_{1i_1}, r_1), \ldots, g_k \in O(r_{k1}, \ldots, r_{ki_k}, r_k)$, an element $f \cdot (g_1, \ldots, g_k) \in O(r_{11}, \ldots, r_{1i_1}, \ldots, r_{k1}, \ldots, r_{ki_k}, r)$
3. for any permutation $\sigma \in \mathbb{R}_k$, a map $\sigma : O(r_1, \ldots, r_k, r) \to O(r_{\sigma(1)}, \ldots, r_{\sigma(k)}, r), f \longmapsto f\sigma$, such that:
   (a) whenever both sides make sense, $f \cdot (g_1 \cdot (h_{11}, \ldots, h_{1i_1}), \ldots, g_k \cdot (h_{k1}, \ldots, h_{ki_k})) = (f \cdot (g_1, \ldots, g_k)) \cdot (h_{11}, \ldots, h_{1i_1}, \ldots, h_{k1}, \ldots, h_{ki_k})$
   (b) for any $f \in O(r_1, \ldots, r_k, r), f = 1_r \cdot f = f \cdot (1_{r_1}, \ldots, 1_{r_k})$
   (c) for any $f \in O(r_1, \ldots, r_k, r)$, and $\sigma, \sigma_1 \in \mathbb{R}_k, f(\sigma\sigma_1) = (f\sigma)\sigma_1$
   (d) for any $f \in O(r_1, \ldots, r_k, r)$, $\sigma \in \mathbb{R}_k$ and $g_1 \in O(r_{11}, \ldots, r_{1i_1}, r_1), \ldots, g_k \in O(r_{k1}, \ldots, r_{ki_k}, r_k), (f\sigma) \cdot (g_{\sigma(1)}, \ldots, g_{\sigma(k)}) = (f \cdot (g_1, \ldots, g_k))\rho(\sigma)$ where $\rho : \mathbb{R}_k \longrightarrow \mathbb{R}_{i_1 + \cdots + i_k}$ is the obvious homomorphism

(e) for any $f \in O(r_1,\dots,r_k,r), g_1 \in O(r_{11},\dots,r_{1i_1},r_1),\dots,g_k \in O(r_{k1},\dots,r_{ki_k},r_k)$, and $\sigma_1 \in \mathbb{R}_{i_1},\dots,\sigma_k \in \mathbb{R}_{i_k}$, $(f\cdot(g_1\sigma_1,\dots,g_k\sigma_k)) = (f\cdot(g_1,\dots,g_k))\varrho_1(\sigma_1,\dots,\sigma_k)$, where $\varrho_1 : \mathbb{R}_{i_1}\times\cdots\times\mathbb{R}_{i_k} \longrightarrow \mathbb{R}_{i_1+\cdots+i_k}$ is the obvious homomorphism.

Thus, an R-operad O can be considered as an algebra where the variables (carrier set) are the typed operators $f \in O(r_1,\dots,r_k,r)$, and the signature is composed by the set of basic operations used for composition of the elements of this carrier set:

1. the binary associative composition operation (a partial function) "$\cdot$," such that $\cdot(f,(g_1,\dots,g_k)) = f\cdot(g_1,\dots,g_k)$, if f is k-ary typed operator,
2. the left and right identity (partial) operations $1_r\cdot_$, $_\cdot(1_{r_1},\dots,1_{r_1})$ such that for a k-ary typed operator f, $1_r\cdot_(f) = 1_r\cdot f = f$, $_\cdot(1_{r_1},\dots,1_{r_1})(f) = f\cdot(1_{r_1},\dots,1_{r_1}) = f$,
3. the permutation (partial) operations, $\sigma : O(r_1,\dots,r_k,r) \to O(r_{\sigma(1)},\dots,r_{\sigma(k)},r)$, such that for each $f \in O(r_1,\dots,r_k,r)$, $\sigma(f) = f\sigma$.

Thus, all operations in the signature of this algebra are the *partial* functions (defined only for the subsets of elements of the carrier set of this algebra).

It is clear, that in this formal syntax of R-operads algebra, the relation symbols are left out of its syntax; thus, the terms of this algebra do not contain the relational symbols. Consequently, it appears fundamentally different from the relational algebras as, e. g., the basic SPRJU Codd's algebra and its extensions.

The theory of typed operads represents the syntax of the database mapping, translated from the schema-database logic into the algebraic framework. The semantic part of the operad's theory, correspondent to the semantics of FOL based on Tarski's interpretations, is represented the R-algebras, which are particular interpretations of the operads.

Let us define now the "R-algebra" of an database mapping-operad, based on homotopy theory [160–162], where its abstract operations are represented by actual functions (query-functions).

Definition 64. For a given universe of values $\mathcal{U}$ and R-operad O, a R-algebra α consists of:

1. for any $r \in \mathbb{R}$, a set $\alpha(r)$ is a set of tuples (relation), with empty relation $\alpha(r_\emptyset) = \bot = \{\langle\rangle\}$ (with empty tuple $\langle\rangle$), unary universe-relation $\alpha(r_\infty) = \{\langle d\rangle \mid d \in \mathcal{U}\}$, and binary relation $\alpha(r_\top) = R_=$ for the "truth" type $r_\top$. The α^* is the extension of α to a list of symbols $\alpha^*(\{r_1,\dots,r_k\}) \triangleq \{\alpha(r_1),\dots,\alpha(r_k),\alpha(r_\emptyset)\}$,
2. for any $q_i \in O(r_1,\dots,r_k,r)$ a mapping function $\alpha(q_i) : R_1\times\cdots\times R_k \longrightarrow \alpha(r)$, where for each $1 \le i \le k$, $R_i = \mathcal{U}^{\mathrm{ar}(r_i)}\backslash\alpha(r_i)$, (here, "\" is a set-difference), if the place symbol $(_)_i \in q_i$ is preceded by negation operator $\neg$; $\alpha(r_i)$ otherwise. Consequently, $\alpha^*(\{q_1,\dots,q_k\}) \triangleq \{\alpha(q_1),\dots,\alpha(q_k)\}$, and
   (a) for any $r \in \mathbb{R}$, $\alpha(1_r)$ acts as an identity function on relation $\alpha(r)$.
   The $q_\bot = \alpha(1_{r_\emptyset}) : \bot \to \bot$ is the empty function (the function with the empty graph)

(b) for the associative composition "$\cdot$" we have that $\alpha(q \cdot (q_1, \ldots, q_k)) = \alpha(q)(\alpha(q_1) \times \cdots \times \alpha(q_k))$
(c) for any $q \in O(r_1, \ldots, r_k, r)$ and a permutation $\sigma \in \mathbb{R}_k$, $\alpha(q\sigma) = \alpha(q)\sigma$, where σ acts on the function $\alpha(q)$ on the right by permuting its arguments

3. we introduce the two functions, ∂_0 and ∂_1, such that for any $\alpha(q)$, $q \in O(r_1, \ldots, r_k, r)$, we have that $\partial_0(q) = \{r_1, \ldots, r_k\}$, $\partial_0(\alpha(q)) = \alpha^*(\partial_0(q)) = \{\alpha(r_1), \ldots, \alpha(r_k)\}$, $\partial_1(q) = \{r\}$ and $\partial_1(\alpha(q)) = \alpha^*(\partial_1(q)) = \{\alpha(r)\}$.

We consider an R-algebra in the next Definition 63 in the way where not the relations but the operads $f \in O(r_1, \ldots, r_k, r)$ constitute its carrier set, and is similar to the consideration of a category as a kind of algebra where the carrier set is composed by objects and arrows, with basic operation just like the associative composition "$\circ'$" of the arrows, which is a partial function, as is the associative operation "$\cdot$" for the operads. This analogy (both with the composition with identity elements) is the natural justification why we are using the interpretation of R-operads algebra in Definition 63 as the basis for the definition of the **DB** category for database mappings.

A.3.4 Sketches and functors into the DB category

We use the algorithm MakeOperads [5] in order to transform logical schema mappings $\mathcal{M}_{AB} = \{\Phi\} : \mathcal{A} \to \mathcal{B}$ given by the SOtgds Φ in Definition 61 into the algebraic operads $\mathbf{M}_{AB} = \text{MakeOperad}(\mathcal{M}_{AB}) = \{v_1 \cdot q_{A,1}, \ldots, v_n \cdot q_{A,1}, 1_{r_\emptyset}\} : \mathcal{A} \to \mathcal{B}$. The basic idea of the operad's operations $v_i \in O(r', r_B)$ and $q_{A,i} \in O(r_1, \ldots, r_k, r')$, where $r_i, 1 \leq i \leq k$ are relational symbols of the source schema $\mathcal{A} = (S_A, \Sigma_A)$ and r_B is a relational symbol of the target schema $\mathcal{B}$, and r' has the same type as r_B, is to formalize algebraically a mapping from the set of source relations r_i into a target relation r_B.

Let us define the algorithm that transform a SOtgd into the set of abstract operads:

Operads algorithm MakeOperads($\mathcal{M}_{AB}$)
Input: A schema mapping $\mathcal{M}_{AB} : \mathcal{A} \to \mathcal{B}$ given by a SOtgd.

$$\exists \mathbf{f}((\forall \mathbf{x}_1(\phi_{A,1} \Rightarrow \psi_{B,1})) \wedge \cdots \wedge (\forall \mathbf{x}_n(\phi_{A,n} \Rightarrow \psi_{B,n}))).$$

Output: Set of abstract operad's operations from $\mathcal{A}$ into $\mathcal{B}$

1. (*Normalize the SOtgd*).
Initialize S to be the empty set $\emptyset$. Let $\mathcal{M}_{AB}$ be the singleton set $\{\exists \mathbf{f}((\forall \mathbf{x}_1(\phi_{A,1} \Rightarrow \psi_{B,1})) \wedge \cdots \wedge (\forall \mathbf{x}_n(\phi_{A,n} \Rightarrow \psi_{B,n})))\}$.
Put each of n implications $\phi_{A,i} \Rightarrow \psi_{B,i}$, for $1 \leq i \leq n$, into S.
Each implication χ in S has the form $\phi_{A,i}(\mathbf{x}_i) \Rightarrow \wedge_1^k r_j(\mathbf{t}_j)$ where every member of $\mathbf{x}_i$ is a universally quantified variable, and each $\mathbf{t}_j$, for $1 \leq j \leq k$, is a sequence (tuple)

of terms over $\mathbf{x}_i$. We then replace each such implication χ in S with k implications: $\phi_{A,i}(\mathbf{x}_i) \Rightarrow r_1(\mathbf{t}_1), \ldots, \phi_{A,i}(\mathbf{x}_i) \Rightarrow r_k(\mathbf{t}_k)$.

2. (*Reinsert the hidden relations*).
In each implication, $\phi_{A,i}(\mathbf{x}_i) \Rightarrow r_k(\mathbf{t}_k)$ in S (obtained in the previous step) in the left-side of this implications replace each equation $(f_r(\mathbf{y}_i) \doteq \overline{1})$, where $\mathbf{y}_i \subseteq \mathbf{x}_i$ and f_r is the characteristic function of a hidden relation $r \notin \mathcal{A}$ (introduced by the algorithm Compose), by the atom $r(\mathbf{y}_i)$.
3. (*Transformation into operad's operations*).
Let $S = \{\chi_1, \ldots, \chi_m\}$ with χ_i a view-based query mapping (implication from conjunctive query over $\mathcal{A}$ into an relation of $\mathcal{B}$), $q_{Ai}(\mathbf{x}_i) \Rightarrow r'(\mathbf{t}_i)$, where $S_q = (r_{i_1,1}, \ldots, r_{i_k,k})$ is the ordered list of relational symbols as they appear (from left to right) in the conjunctive query $q_{Ai}(\mathbf{x}_i)$. We then replace each such implication χ_i in S with the operad's operations $q_i \in O(r_{i_1,1}, \ldots, r_{i_k,k}, r')$ where $(r_{i_1,1}, \ldots, r_{i_k,k}) = S_q$ and q_i is the expression obtained from $q_{Ai}(\mathbf{x}_i)$ by replacing each relational symbol $r_{i_j,j}$ by the place symbol $(_)_j$ and by replacing r' by unlabeled $(_)$.
We will represent each operation q_i by a composition of two operations $v_i \cdot q_{A,i}$, where $q_{A,i} \in O(r_{i_1,1}, \ldots, r_{i_k,k}, r_q)$ is the same expression as q_i above where r_q is a relation symbol of the same type as r', and $v_i \in O(r_q, r')$ is the expression $(_)_1(\mathbf{y}_i) \Rightarrow (_)(\mathbf{y}_i)$ where $\mathbf{y}_i$ is the tuple of variables of the atoms $r'(\mathbf{y}_i)$ and $r_q(\mathbf{y}_i)$.
In the simplest case when χ_i is a tautology $r(\mathbf{x}_i) \Rightarrow r(\mathbf{x}_i)$, then χ_i is replaced in S by $q_i = q_{A,i} = v_i = 1_r \in O(r, r)$ (i. e., by identity mapping expression $(_)_1(\mathbf{x}_i) \Rightarrow (_)(\mathbf{x}_i)$ if $\mathbf{x}_i$ is not empty; the expression $(_)_1 \Rightarrow (_)$ for identity operation $1_{r_\emptyset} \in O(r_\emptyset, r_\emptyset)$, otherwise).
4. (*Construct operad's operations*).
Add the "empty operation" operad $1_{r_\emptyset} \in O(r_\emptyset, r_\emptyset)$ in S, represented by the mapping expression $(_)_1 \Rightarrow (_)$.
Return The set of abstract operad's operations denoted by an (mapping) arrow $\mathbf{M}_{AB} = S : \mathcal{A} \to \mathcal{B}$.

In what follows, we will use the term "mapping operad" from such a set of abstract operad's operations. The reason that we insert the identity "empty operad's operation" $1_{r_\emptyset} \in O(r_\emptyset, r_\emptyset)$ in $\mathbf{M}_{AB} = S$ is that, as we will see, we can have the Tarski's interpretations of database mappings such that we do not transfer any information from source to target database (with the empty information flux), and this empty operad's operation has empty information flux. Consequently, the $1_{r_\emptyset}$ is always an element of the set of operad's operations obtained from a given SOtgd. As we will see in the algorithm of decomposition of SOtgds, we will have the cases when there are generated mappings with SOtgd equal to the tautology $r_\emptyset \Rightarrow r_\emptyset$.

Example 29. In the most trivial case when $\mathcal{M}_{AB} = \{r_\emptyset \Rightarrow r_\emptyset\}$ (i. e., when there is not an effective mapping between the schemas $\mathcal{A}$ and $\mathcal{B}$), where $r_\emptyset$ is a nullary predicate symbol (i. e., an propositional letter), so that SOtgd is a banal tautology $r_\emptyset \Rightarrow r_\emptyset$ and,

consequently, $\mathcal{M}_{AB}$ is always satisfied; we obtain that MakeOperads($\mathcal{M}_{AB}$) = $\{1_{r_\emptyset}\}$. The particular cases for such trivial mappings are the following:

$$\mathcal{M}_{AA_\emptyset} = \{r_\emptyset \Rightarrow r_\emptyset\} : \mathcal{A} \to \mathcal{A}_\emptyset; \quad \mathcal{M}_{A_\emptyset A} = \{r_\emptyset \Rightarrow r_\emptyset\} : \mathcal{A}_\emptyset \to \mathcal{A}; \quad \text{and}$$
$$\mathrm{Id}_{A_\emptyset} = \{r_\emptyset \Rightarrow r_\emptyset\} : \mathcal{A}_\emptyset \to \mathcal{A}_\emptyset,$$

the identity mapping, where $\mathcal{A}_\emptyset = (\emptyset, \emptyset)$ is the empty database schema. It is easy to verify that MakeOperads($\{r_\emptyset \Rightarrow r_\emptyset\}$) = $\{1_{r_\emptyset}\}$.

Obviously, we cannot have the Tarski's interpretations for empty schema $\mathcal{A}_\emptyset$, and we will need to define their interpretation in an ad hoc way in Section A.3.5 dedicated to R-algebras for the operads.

There is an obvious inverse transformation of a given mapping-operad (a set of k-ary operations) $\mathbf{M}_{AB} = \{q_1, \ldots, q_n, 1_{r_\emptyset}\} : \mathcal{A} \to \mathcal{B}$, into the SOtgd of this mapping. Let us define the algorithm that transform a set of mapping operads into the SOtgd:

Inverse operads algorithm InverseOperads($\mathbf{M}_{AB}$)
Input: A mapping-operad $\mathbf{M}_{AB} = \{q_1, \ldots, q_n, 1_{r_\emptyset}\} : \mathcal{A} \to \mathcal{B}$.
Output: Mapping $\mathcal{M}_{AB}$ from $\mathcal{A}$ into $\mathcal{B}$ represented by the SOtgd.

1. (*Transform operads into logic formulae*).
   Initialize S to be the empty set $\emptyset$.
   For each k-ary mapping operad's operation $q_i \in O(r_{i,1}, \ldots, r_{i,k}, r')$, for $1 \le i \le n$, such that $q_i \neq 1_{r_\emptyset}$, add in S the logic sentence $\forall \mathbf{x}_i \phi_i$ where $\mathbf{x}_i$ is the tuple of all variables in the expression of the operad's operation q_i and ϕ_i is the logic formula where each labeled place symbol $(_)_m$, for $1 \le m \le k$, is replaced by the relation $r_{i,k}$ and unlabeled place symbol $(_)$ is replaced by $r' \in \mathcal{B}$.
2. (*Elimination of relation symbols not in $\mathcal{A}$*).
   In each formula $\forall \mathbf{x}_i \phi_i \in S$ where ϕ is an implication $\phi_{A,i}(\mathbf{x}_i) \Rightarrow r'(\mathbf{t}_i)$, in the left-side of these implications replace each atom $r(\mathbf{y}_i)$ with $r \notin \mathcal{A}$, where $\mathbf{y}_i \subseteq \mathbf{x}_i$, by the equation $(f_r(\mathbf{y}_i) \doteq \bar{1})$, where f_r is the characteristic function of a hidden relation $r \notin \mathcal{A}$.
3. (*Construct* SOtgd).
   The SOtgd is a formula Φ equal to $\exists \mathbf{f}(\forall \mathbf{x}_1 \phi_1 \wedge \cdots \wedge \forall \mathbf{x}_n \phi_n)$, where $\mathbf{f}$ is the tuple of all functional symbols in the formulae in S.
   Return the logic mapping $\mathcal{M}_{AB} = \{\Phi\} : \mathcal{A} \to \mathcal{B}$.

Consequently, the algebraic formalism based on the mapping operads is equivalent to the logic formalism based on SOtgds.

Example 30. Schema $\mathcal{A} = (S_A, \emptyset)$ consists of a unary relation `EmpAcme` that represents the employees of Acme, a unary relation `EmpAjax` that represents the employees of Ajax, and unary relation `Local` that represents employees that work in the local office of their

company. Schema $\mathcal{B} = (S_B, \emptyset)$ consists of a unary relation Emp that represents all employees, a unary relation Local1 that is intended to be a copy of Local and unary relation Over65 that is intended to represent people over age 65. Schema $\mathcal{C} = (S_C, \emptyset)$ consists of a binary relation Office that associates employees with office numbers and unary relation CanRetire that represents employees eligible for retirement. Consider now the following schema mappings:

$$\begin{aligned}\mathcal{M}_{AB} = \{&\forall x_e(\mathtt{EmpAcme}(x_e) \Rightarrow \mathtt{Emp}(x_e)) \wedge \forall x_e(\mathtt{EmpAjax}(x_e) \Rightarrow \mathtt{Emp}(x_e))\\ &\wedge \forall x_p(\mathtt{Local}(x_p) \Rightarrow \mathtt{Local1}(x_p))\}, \quad \text{and}\\ \mathcal{M}_{BC} = \{&\exists f_1(\forall x_e((\mathtt{Emp}(x_e) \wedge \mathtt{Local1}(x_e)) \Rightarrow \mathtt{Office}(x_e, f_1(x_e)))\\ &\wedge \forall x_e((\mathtt{Emp}(x_e) \wedge \mathtt{Over65}(x_e)) \Rightarrow \mathtt{CanRetire}(x_e)))\}.\end{aligned}$$

Then, by their composition, we obtain the composed mapping $\mathcal{M}_{AC} : \mathcal{A} \to \mathcal{C}$ equal to

$$\begin{aligned}\mathcal{M}_{AC} = \{&\exists f_1 \exists f_2 \exists f_{\text{Over65}}(\forall x_e((\mathtt{EmpAcme}(x_e) \wedge \mathtt{Local}(x_e)) \Rightarrow \mathtt{Office}(x_e, f_1(x_e)))\\ &\wedge \forall x_e((\mathtt{EmpAjax}(x_e) \wedge \mathtt{Local}(x_e)) \Rightarrow \mathtt{Office}(x_e, f_2(x_e)))\\ &\wedge \forall x_e((\mathtt{EmpAcme}(x_e) \wedge (f_{\text{Over65}}(x_e) \doteq \bar{1})) \Rightarrow \mathtt{CanRetire}(x_e))\\ &\wedge \forall x_e((\mathtt{EmpAjax}(x_e) \wedge (f_{\text{Over65}}(x_e) \doteq \bar{1})) \Rightarrow \mathtt{CanRetire}(x_e)))\},\end{aligned}$$

where f_{Over65} is the characteristic function of the relation (predicate) Over65 which is not part of schema $\mathcal{A}$. Then, by transformation into abstract operad's operations, we obtain $\mathbf{M}_{AC} = \text{MakeOperads}(\mathcal{M}_{AC}) = \{q_1^A, q_2^A, q_3^A, q_4^A, 1_{r_\emptyset}\}$, $q_i^A = v_i \cdot q_{A,i}$, where:

1. The operations $q_1^A \in O(\mathtt{EmpAcme}, \mathtt{Local}, \mathtt{Office})$ and $q_{A,1} \in O(\mathtt{EmpAcme}, \mathtt{Local}, r_1')$ correspond to the expression $((_)_1(x_e) \wedge (_)_2(x_e)) \Rightarrow (_)(x_e, f_1(x_e))$ and $v_1 \in O(r_1', \mathtt{Office})$ to $(_)_1(x_e, x_p) \Rightarrow (_)(x_e, x_p)$;
2. The operations $q_2^A \in O(\mathtt{EmpAjax}, \mathtt{Local}, \mathtt{Office})$ and $q_{A,2} \in O(\mathtt{EmpAjax}, \mathtt{Local}, r_2')$ correspond to the expression $((_)_1(x_e) \wedge (_)_2(x_e)) \Rightarrow (_)(x_e, f_2(x_e))$ and $v_2 \in O(r_2', \mathtt{Office})$ to $(_)_1(x_e, x_p) \Rightarrow (_)(x_e, x_p)$;
3. The operations $q_3^A \in O(\mathtt{EmpAcme}, \mathtt{Over65}, \mathtt{CanRetire})$ and $q_{A,3} \in O(\mathtt{EmpAcme}, \mathtt{Over65}, r_3')$ correspond to the expression $((_)_1(x_e) \wedge (_)_2(x_e)) \Rightarrow (_)(x_e)$ and $v_3 \in O(r_3', \mathtt{CanRetire})$ to $(_)_1(x_e) \Rightarrow (_)(x_e)$;
4. The operations $q_4^A \in O(\mathtt{EmpAjax}, \mathtt{Over65}, \mathtt{CanRetire})$ and $q_{A,4} \in O(\mathtt{EmpAjax}, \mathtt{Over65}, r_4')$ correspond to the expression $((_)_1(x_e) \wedge (_)_2(x_e)) \Rightarrow (_)(x_e)$ and $v_4 \in O(r_4', \mathtt{CanRetire})$ to $(_)_1(x_e) \Rightarrow (_)(x_e)$.

Based on the equivalence of representation of schema mappings by SOtgd and by mapping-operads, we can introduce the composition of mapping operads analogously to the composition of schema mappings.

Definition 65. For any given schema mappings $\mathcal{M}_{AB} : \mathcal{A} \to \mathcal{B}$ and an *atomic* schema mapping $\mathcal{M}_{BC} : \mathcal{B} \to \mathcal{C}$, we can define the correspondent mapping-operads $\mathbf{M}_{AB} =$

MakeOperads($\mathcal{M}_{AB}$) = $\{q_1^A, \ldots, q_n^A, 1_{r_\emptyset}\} : \mathcal{A} \to \mathcal{B}$ and $\mathbf{M}_{BC}$ = MakeOperads($\mathcal{M}_{BC}$) = $\{q_1^B, \ldots, q_m^B, 1_{r_\emptyset}\} : \mathcal{B} \to \mathcal{C}$, and their composition $\mathbf{M}_{BC} \circ \mathbf{M}_{AB} \triangleq \{1_{r_\emptyset}\} \bigcup_{1 \le i \le m} \{q_i = q_i^B \cdot (q_{i1}^A, \ldots, q_{ij}^A) \mid$ where $q_i^B \in O(r_{B,i1}, \ldots, r_{B,ij}, r_{C,i}) \in \mathbf{M}_{BC}$, and $(q_{ik}^A \in O(r_{k1}, \ldots, r_{kl}, r_{B,ik})$ if $q_{ik}^A \in \mathbf{M}_{AB}$; $q_{ik}^A = 1_{r_{B,ik}}$ otherwise $)_{1 \le k \le j}\}$.

So, these three arrows $\mathbf{M}_{AB} : \mathcal{A} \to \mathcal{B}, \mathbf{M}_{BC} : \mathcal{B} \to \mathcal{C}$ and $\mathbf{M}_{AC} : \mathcal{A} \to \mathcal{C}$, in the example above, compose a graph G of this database mapping system.

From the fact that the operads can be composed, the composition of two schema mappings $\mathbf{M}_{AB}$ and $\mathbf{M}_{BC}$ can be translated into composition of operads, which is associative, so that they can be represented by the sketch category **Sch**(G) derived from the graph G of the schema mappings. It was demonstrated [5] that the transformation of SOtgds of the schema mappings into mapping operads by the algorithm MakeOperads is well-defined and satisfy the general properties of operads in Definition 63.

Sketches are developed by Ehresmann's school, especially by R. Guitartand and C. Lair [110, 164, 165]. Sketches are called graph-based logic and provide very clear and intuitive specification of computational data and activities. For any small sketch **E**, the category of models Mod(**E**) is an accessible category by Lair's theorem and reflexive subcategory of $\mathbf{Set}^{\mathbf{E}}$ by the Ehresmann–Kennison theorem. A generalization to base categories other than **Set** was proved by Freyd and Kelly (1972) [166]. The generalization to **DB** category (see Theorem 10 in Section 7.1) is exhaustively provided in [5], so that the functorial semantics of a database mapping system expressed by a graph G is defined by a functor (R-algebra) $\alpha^* : \mathbf{Sch}(G) \to \mathbf{DB}$.

Sketch is a category together with a distinguished class of cones and cocones. A *model* of the sketch is a set-valued functor turning all distinguished cones into limit cones, and all distinguished cocones into colimit cocones, in the category **Set** of sets.

The category of models of a given sketch has models as objects and the arrows that represent all natural transformations between the models as functors. A category is sketchable (esquissable) or accessible iff it is equivalent to the category of set-valued models of a small sketch.

Recall that a graph $G = (V_G, E_G)$ consists of a set of vertices denoted by V_G and a set of arrows (edges) denoted by E_G together with the operators $\mathrm{dom}, \mathrm{cod} : E_G \to V_G$, which assigns to each arrow its source and target. (Co)cones and diagrams are defined for graphs in exactly the same way as they are for categories, but commutative co(cones) and diagrams, of course, make no sense for graphs.

By a sketch we mean 4-tuple (G, u, D, C) where G is a graph, $u : V_G \to E_G$ is a function, which takes each vertex (node) $\mathcal{A}$ in V_G to an arrow from $\mathcal{A}$ to $\mathcal{A}$, D is a class of diagrams in G and C is a class of (co)cones in G. Each (c)cone in G goes (to)from some vertex (from)to some diagram; that diagram need not be in D and, in fact, it is necessary to allow diagrams, which are not in D as bases of (co)cones.

The R-algebra α is derived from a given Tarski's interpretation I_T of the given database schema mapping graph G and represented by a sketch category **Sch**(G) (with arrows $M_{AB} : \mathcal{A} \to \mathcal{B}$, as in Example 30). R-algebra α is equal to I_T for the

relations of the data schemas, $\alpha(r_i) = I_T(r_i)$ is a relational table of the instance database $A = \alpha^*(\mathcal{A}) = \{\alpha(r_i) \mid r_i \in S_A\}$ (α^* denotes the extension of α to sets), and $\alpha(q_{A,i}) : \alpha(r_1) \times \cdots \times \alpha(r_k) \to \alpha(r')$ is a surjective function from the relations in the instance database A into its image (relation) $\alpha(r')$, with a function $\alpha(v_i) : \alpha(r') \to \alpha(r_B)$ into the relation of the instance database $B = \alpha^*(\mathcal{B})$. Formal definition of an R-algebra α as a mapping interpretation of a schema mapping $\mathcal{M}_{AB} : \mathcal{A} \to \mathcal{B}$ is given by Definition 66 also when we have the negation in the left sides of the implications in a SOtgd of such a mapping.

We have that for any R-algebra α, $\alpha(r_\emptyset) = \bot = \{\langle\rangle\}$ is the empty relation composed by only empty tuple $\langle\rangle \in D_{-1}$, and $1_{r_\emptyset}$ is the identity operads operation of the empty relation $r_\emptyset$, so that $q_\bot = \alpha(1_{r_\emptyset}) = \mathrm{id}_\bot : \bot \to \bot$ is the identity function.

Example 31. For the operads defined in Example 30, let a mapping interpretation (an R-algebra) α be an extension of Tarski's interpretation I_T of the source schema $\mathcal{A} = (S_A, \Sigma_A)$ that satisfies all constraints in Σ_A and defines its database instance $A = \alpha^*(S_A) = \{\alpha(r_i) \mid r_i \in S_A\}$ and, analogously, an interpretation of $\mathcal{C}$.

Let α satisfy the SOtgd of the mapping $\mathcal{M}_{AC}$ by the Tarski's interpretation for the functional symbols f_i, for $1 \le i \le 2$, in this SOtgd (denoted by $I_T(f_i)$).

Then we obtain the relations $\alpha(\texttt{EmpAcme})$, $\alpha(\texttt{EmpAjax})$, $\alpha(\texttt{Local})$, $\alpha(\texttt{Office})$ and $\alpha(\texttt{CanRetire})$. The interpretation of f_{Over65} is the characteristic function of the relation $\alpha(\texttt{Over65}$ in the instance $B = \alpha^*(S_B)$ of the database $\mathcal{B} = (S_B, \Sigma_B)$, so that $\overline{f}_{\text{Over65}}(a) = 1$ if $\langle a\rangle \in \alpha(\texttt{Over65}))$.

Then this mapping interpretation α defines the following functions:

1. The function $\alpha(q_{A,1}) : \alpha(\texttt{EmpAcme}) \times \alpha(\texttt{Local}) \to \alpha(r_1')$, such that for any tuple $\langle a\rangle \in \alpha(\texttt{EmpAcme})$ and $\langle b\rangle \in \alpha(\texttt{Local})$,

$$\alpha(q_{A,1})(\langle a\rangle, \langle b\rangle) = \langle a, I_T(f_1(a))\rangle \quad \text{if } a = b; \quad \langle\rangle \text{ otherwise.}$$

   And for any $\langle a, b\rangle \in \alpha(r_1')$, $\alpha(v_1)(\langle a, b\rangle) = \langle a, b\rangle$ if $\langle a, b\rangle \in \alpha(\texttt{Office})$; $\langle\rangle$ otherwise.
2. The function $\alpha(q_{A,2}) : \alpha(\texttt{EmpAjax}) \times \alpha(\texttt{Local}) \to \alpha(r_2')$, such that for any tuple $\langle a\rangle \in \alpha(\texttt{EmpAjax})$ and $\langle b\rangle \in \alpha(\texttt{Local})$,

$$\alpha(q_{A,2})(\langle a\rangle, \langle b\rangle) = \langle a, I_T(f_2(a))\rangle \quad \text{if } a = b; \quad \langle\rangle \text{ otherwise.}$$

   And for any $\langle a, b\rangle \in \alpha(r_2')$, $\alpha(v_2)(\langle a, b\rangle) = \langle a, b\rangle$ if $\langle a, b\rangle \in \alpha(\texttt{Office})$; $\langle\rangle$ otherwise.
3. The function $\alpha(q_{A,3}) : \alpha(\texttt{EmpAcme}) \times \alpha(\texttt{Over65}) \to \alpha(r_3')$, such that for any tuple $\langle a\rangle \in \alpha(\texttt{EmpAcme})$ and $\langle b\rangle \in \alpha(\texttt{Over65})$,

$$\alpha(q_{A,3})(\langle a\rangle, \langle b\rangle) = \langle a\rangle, \quad \text{if } a = b; \quad \langle\rangle \text{ otherwise.}$$

   And for any $\langle a\rangle \in \alpha(r_3')$, $\alpha(v_3)(\langle a\rangle) = \langle a\rangle$ if $\langle a\rangle \in \alpha(\texttt{CanRetire})$; $\langle\rangle$ otherwise.
4. The function $\alpha(q_{A,4}) : \alpha(\texttt{EmpAjax}) \times \alpha(\texttt{Over65}) \to \alpha(r_4')$, such that for any tuple $\langle a\rangle \in \alpha(\texttt{EmpAjax})$ and $\langle b\rangle \in \alpha(\texttt{Over65})$

$$\alpha(q_{A,4})(\langle a\rangle, \langle b\rangle) = \langle a\rangle, \quad \text{if } a = b; \quad \langle\rangle \text{ otherwise.}$$

And for any $\langle a\rangle \in \alpha(r'_4)$, $\alpha(v_4)(\langle a\rangle) = \langle a\rangle$ if $\langle a\rangle \in \alpha(\texttt{CanRetire})$; $\langle\rangle$ otherwise.

From the fact that the mapping interpretation satisfies the schema mappings, based on Corollary 4 in Section 2.4.1 [5], all functions $\alpha(v_i)$, for $1 \le i \le 4$, are the injections.

The categorial logic used for the denotational semantics in data integration theory [5] is based on the category **DB** (see Theorem 10 in Section 7.1) where the objects are the instances of RDBs and the morphisms are the set of functions, which express the schema mappings between RDBs. It was demonstrated that the denotational semantics of database mappings is given by morphisms of the Kleisly category given by a power-view monad over the category **DB**.

The notion of a monad is one of the most general mathematical notions. For instance, *every* algebraic theory, i. e., every set of operations satisfying equational laws, can be seen as a monad. Moggi [94, 95] stressed the computational significance of monads and explained how they may help understand programs "*as functions from values to computations.*" The idea of Moggi, roughly, is to give denotational semantics to computations and it presents an alternative to the conceptual gap between the intensional (operational) and the extensional (denotational) approaches to the semantics of programming languages.

The idea of monad as a model for computations, based on an endofunctor T for a given category, is that for each set of values of type A, TA is the object of computations of "type A," introduced in Section A.3.2. Let us explain the way we can use such denotational semantics, based on monads, in the case of relational databases. It is well known that the relational databases are complex structures, defined by some sets of n-ary relations. In addition, the mappings between the relational databases are based on some sets of view mappings between a source database A and a target database B, as specified in Section A.3.4. We consider the views as an universal property for databases (i. e., possible observations of the information contained in some database).

A categorical logic (denotational semantics) for database schema mapping *based on views*, presented in [5], is a very general framework for database-integration/exchange and peer-to-peer. The base database category **DB** (instead of traditional **Set** category of sets and functions between them), with objects instance databases and with morphisms (mappings which are not simple functions) between them, is used at an *instance level* as a proper semantic domain for a database mappings based on a set of complex query computations.

The higher logical *schema level* of mappings between databases, usually written in some high expressive logical language (e. g., [92, 99], GLAV (LAV and GAV), tuple generating dependency) can then be translated functorially into this base "computation"

category. We will use such a framework also because a Relational DB (RDB) is formally defined as a GAV Data Integration system, so we are able to use all mathematical tools developed in [5] in order to present a formal denotational semantics for the RDBs.

A similar approach in using Kleisli semantics to relax the atomicity requirement for data in standard RDBs has been recently presented in [167], but in that case the level of presentation of a database schema is given by the graphs (sketch categories) composed by binary relations only and the functional dependencies between them. Consequently, in such a low level schema representation we can use the topos **Set** category as a base category and simple powerset functor as a monad, while in our higher-level abstraction we present the whole RDB schema as a node of the sketch and we represent also complex integrity constraints over schemas by schema mappings. Consequently, in our higher level presentation framework the monads have to be developed for the weak monoidal topos **DB**, which is a somewhat mathematically more complicated task.

A.3.5 Semantics of DB schema mappings: Information fluxes

Formal definition of an R-algebra α given by Definition 64 as a mapping interpretation of a schema mapping $\mathcal{M}_{AB} : \mathcal{A} \to \mathcal{B}$ can be reformulated as follows.

Definition 66. Let $\phi_{Ai}(\mathbf{x}) \Rightarrow r_B(\mathbf{t})$ be an implication χ in a normalized SOtgd $\exists \mathbf{f}(\Psi)$ (where Ψ is a FOL formula) of the mapping $\mathcal{M}_{AB}$, $\mathbf{t}$ be a tuple of terms with variables in $\mathbf{x} = (x_1, \ldots, x_m)$, and $q_i \in \text{MakeOperads}(\mathcal{M}_{AB})$ be the operad's operation of this implication obtained by MakeOperads algorithm, equal to the expression $(e \Rightarrow (_)(\mathbf{t})) \in O(r_1, \ldots, r_k, r_B)$, where $q_i = v_i \cdot q_{A,i}$ with $q_{A,i} \in O(r_1, \ldots, r_k, r_q)$ and $v_i \in O(r_q, r_B)$ such that for a new relational symbol r_q, $\text{ar}(r_q) = \text{ar}(r_B) \geq 1$.

Let S be an empty set and $e[(_)_n/r_n]_{1 \leq n \leq k}$ be the formula obtained from expression e where each place-symbol $(_)_n$ is substituted by relational symbol r_n for $1 \leq n \leq k$. Then do the following as far as it is possible: For each two relational symbols r_j, r_n in the formula $e[(_)_n/r_n]_{1 \leq n \leq k}$ such that j_h-th *free variable* (which is not an argument of a functional symbol) in the atom $r_j(\mathbf{t}_j)$ is equal to n_h-th free variable in the atom $r_n(\mathbf{t}_n)$ (both atoms in $e[(_)_n/r_n]_{1 \leq n \leq k}$), we insert the set $\{(j_h, j), (n_h, n)\}$ as one element of S. At the end, S is the set of sets that contain the pairs of mutually equal free variables.

An R-algebra α is a *mapping interpretation* of $\mathcal{M}_{AB} : \mathcal{A} \to \mathcal{B}$ if it is an extension of a Tarski's interpretation I_T, of all predicate and functional symbols in FOL formula Ψ, with I_T^* being its extension (A.1) to all formulae, and if for each $q_i \in \text{MakeOperads}(\mathcal{M}_{AB})$ it satisfies the following:

1. For each relational symbol $r_i \neq r_\emptyset$ in $\mathcal{A}$ or $\mathcal{B}$, $\alpha(r_i) = I_T(r_i)$.
2. We obtain a function $f = \alpha(q_{A,i}) : R_1 \times \cdots \times R_k \to \alpha(r_q)$,

where for each $1 \leq i \leq k$, $R_i = \mathcal{D}^{\mathrm{ar}(r_i)} \backslash \alpha(r_i)$ if the place symbol $(_)_i \in q_i$ is preceded by negation operator $\neg$; $\alpha(r_i)$ otherwise, such that for every $\mathbf{d}_i \in R_i$:

$$f(\mathbf{d}_1, \ldots, \mathbf{d}_k) = g^*(\mathbf{t}) = \big(g^*(t_1), \ldots, g^*(t_{\mathrm{ar}(r_B)})\big)$$

if $\bigwedge\{\pi_{j_h}(\mathbf{d}_j) = \pi_{n_h}(\mathbf{d}_n) \mid \{(j_h, j), (n_h, n)\} \in S\}$ is true and the assignment g satisfies the formula $e[(_)_n/r_n]_{1\leq n\leq k}$; $\langle\rangle$ (empty tuple) otherwise, where the assignment $g : \{x_1, \ldots, x_m\} \to \mathcal{D}$ is defined by the tuple of values $(g(x_1), \ldots, g(x_m)) = \mathrm{Cmp}(S, (\mathbf{d}_1, \ldots, \mathbf{d}_k))$, and its extension g^* to all terms such that for any term $f_i(t_1, \ldots, t_n)$:

$$g^*(f_i(t_1, \ldots, t_n)) = I_T(f_i)(g^*(t_1), \ldots, g^*(t_n)) \quad \text{if } n \geq 1;\ I_T(f_i) \text{ otherwise.}$$

The algorithm Cmp (compacting the list of tuples by eliminating the duplicates defined in S) is defined as follows:

Input: a set S of joined (equal) variables defined above, and a list of tuples $(\mathbf{d}_1, \ldots, \mathbf{d}_k)$.

Initialize $\mathbf{d}$ to $\mathbf{d}_1$. Repeat consecutively the following, for $j = 2, \ldots, k$:

Let $\mathbf{d}_j$ by a tuple of values $(v_1, \ldots, v_{j_n})$, then for $i = 1, \ldots, j_n$ repeat consecutively the following:

$\mathbf{d} = \mathbf{d}\&v_i$ if there does not exist and element $\{(j_h, j), (n_h, n)\}$ in S such that $j \leq n$; $\mathbf{d}$, otherwise.

(The operation of concatenation $'\&'$ appends the value v_i at the end of tuple $\mathbf{d}$)

Output: The tuple $\mathrm{Cmp}(S, (\mathbf{d}_1, \ldots, \mathbf{d}_k)) = \mathbf{d}$.

3. $\alpha(r_q)$ is equal to the image of the function f in point 2 above.
4. The function $h = \alpha(v_i) : \alpha(r_q) \to \alpha(r_B)$ such that for each $\mathbf{b} \in \alpha(r_q)$,

$$h(\mathbf{b}) = \mathbf{b} \quad \text{if } \mathbf{b} \in \alpha(r_B);\ \text{empty tuple } \langle\rangle \text{ otherwise.}$$

Note that the formulae $\phi_{Ai}(\mathbf{x})$ and expression $e[(_)_n/r_n]_{1\leq n\leq k}$ are logically equivalent, with the only difference that the atoms with characteristic functions $f_r(\mathbf{t}) = 1$ in the first formula are substituted by the atoms $r(\mathbf{t})$, based on the fact that the assignment g satisfies $r(\mathbf{t})$ iff $g^*(f_r(\mathbf{t})) = f_r(g^*(\mathbf{t})) = 1$, where $f_r : \mathcal{D}^{\mathrm{ar}(r)} \to \{0, 1\}$ is the characteristic function of relation $\alpha(r)$ such that for each tuple $\mathbf{c} \in \mathcal{D}^{\mathrm{ar}(r)}$, $f_r(\mathbf{c}) = 1$ if $\mathbf{c} \in \alpha(r)$; 0 otherwise.

Example 32. Let us show how we construct the set S and the compacting of tuples given by Definition 66 above.

Let us consider an operad $q_i \in \mathrm{MakeOperads}(\mathcal{M}_{AB})$, obtained from a normalized implication $\phi_{Ai}(\mathbf{x}) \Rightarrow r_B(\mathbf{t})$ in $\mathcal{M}_{AB}$, $((y = f_1(x, z)) \wedge r_1(x, y, z) \wedge r_2(v, x, w) \wedge (f_{r_3}(y, z, w', w) = 1)) \Rightarrow r_B(x, z, w, f_2(v, z))$, so that q_i is equal to the expression $(e \Rightarrow (_)(\mathbf{t})) \in O(r_1, r_2, r_3, r_B)$, where $\mathbf{x} = (x, y, z, v, w, w')$ (the ordering of variables in the atoms (with database relational symbols) from left to right), $\mathbf{t} = (x, z, w, f_2(v, z))$ and

the expression e equal to $(y \doteq f_1(x,z)) \wedge (_)_1(\mathbf{t}_1) \wedge (_)_2(\mathbf{t}_2) \wedge (_)_3(\mathbf{t}_3)$, with $\mathbf{t}_1 = (x,y,z)$, $\mathbf{t}_2 = (v,x,w)$ and $\mathbf{t}_3 = (y,z,w',w)$. Consequently, we obtain

$$S = \{\{(1,1),(2,2)\},\{(2,1),(1,3)\},\{(3,1),(2,3)\},\{(3,2),(4,3)\}\},$$

that are the positions of duplicates (or joined variables) of x, y, z and w, respectively.

Thus, for given tuples $\mathbf{d}_1 = (a_1,a_2,a_3) \in \alpha(r_1)$, $\mathbf{d}_2 = (b_1,b_2,b_3) \in \alpha(r_2)$ and $\mathbf{d}_3 = (c_1,c_2,c_3,c_4) \in \alpha(r_3)$, the statement $\bigwedge\{\pi_{j_h}(\mathbf{d}_j) = \pi_{n_h}(\mathbf{d}_n) \mid \{(j_h,j),(n_h,n)\} \in S\}$ is equal to $(\pi_1(\mathbf{d}_1) = \pi_2(\mathbf{d}_2)) \wedge (\pi_2(\mathbf{d}_1) = \pi_1(\mathbf{d}_3)) \wedge (\pi_3(\mathbf{d}_1) = \pi_2(\mathbf{d}_3)) \wedge (\pi_3(\mathbf{d}_2) = \pi_4(\mathbf{d}_3))$, which is true when $a_1 = b_2$, $a_2 = c_1$, $a_3 = c_2$ and $b_3 = c_4$.

The compacting of these tuples is equal to

$$\mathbf{d} = \mathrm{Cmp}(S,(\mathbf{d}_1,\mathbf{d}_2,\mathbf{d}_3)) = (a_1,a_2,a_3,b_1,b_3,c_3),$$

with the assignment to variables $[x/a_1]$, $[y/a_2]$, $[z/a_3]$, $[v,b_1]$, $[w/b_3]$ and $[w'/c_3]$.

That is, $\mathbf{d} = \mathbf{x}[x/a_1, y/a_2, z/a_3, v/b_1, w/b_3, w'/c_3]$ is obtained by this assignment g to the tuple of variables $\mathbf{x}$, so that the sentence $e[(_)_n/r_n]_{1\leq n\leq k}/g$ is well-defined and equal to

$$(a_2 = I_T(f_1)(a_1,a_3)) \wedge r_1(a_1,a_2,a_3) \wedge r_2(b_1,a_1,b_3) \wedge r_3(a_2,a_3,c_3,b_3), \quad \text{i. e., to}$$
$$(a_2 = I_T(f_1)(a_1,a_3)) \wedge r_1(\mathbf{d}_1) \wedge r_2(\mathbf{d}_2) \wedge r_3(\mathbf{d}_3),$$

and if this formula is satisfied by such an assignment g, i. e., $I_T^*(e[(_)_n/r_n]_{1\leq n\leq k}/g) = 1$, then

$$\begin{aligned} f((\mathbf{d}_1,\mathbf{d}_2,\mathbf{d}_3)) = g^*(\mathbf{t}) &= (g(x),g(z),g(w),g^*(f_2(v,z))) \\ &= (a_1,a_3,b_3,I_T(f_2)(b_1,a_3)), \end{aligned}$$

for a given Tarski's interpretation I_T, where I_T^* is the extension of I_T to all FOL formulae.

If $\mathcal{M}_{AB}$ is satisfied by the mapping interpretation α, this value of $f((\mathbf{d}_1,\mathbf{d}_2,\mathbf{d}_3))$ corresponds to the truth of the normalized implication in the SOtgd of $\mathcal{M}_{AB}$, $\phi_{Ai}(\mathbf{x}) \Rightarrow r_B(\mathbf{t})$ for the assignment g derived by substitution $[\mathbf{x}/\mathbf{d}]$, when $\phi_{Ai}(\mathbf{x})/g$ is true. Hence, $r_B(\mathbf{t})/g$ is equal to $r_B(a_1,a_3,b_3,I_T(f_2)(b_1,a_3))$, i. e., to $r_B(f((\mathbf{d}_1,\mathbf{d}_2,\mathbf{d}_3)))$ and has to be true as well (i. e., $I_T^*(r_B(f((\mathbf{d}_1,\mathbf{d}_2,\mathbf{d}_3))))) = 1$ or, equivalently, $f((\mathbf{d}_1,\mathbf{d}_2,\mathbf{d}_3)) \in \alpha(r_B) = I_T(r_B)$).

Consequently, if $\mathcal{M}_{AB}$ is satisfied by a mapping interpretation α (and hence $\alpha(v_i)$ is an injection function with $\alpha(r_q) \subseteq \alpha(r_B)$), then $f((\mathbf{d}_1,\mathbf{d}_2,\mathbf{d}_3)) \in \|r_B\|$, so that the function $f = \alpha(q_{A,i})$ represents the transferring of the tuples in relations of the source instance databases into the target instance database $B = \alpha^*(\mathcal{B})$, according to the SOtgd Φ of the mapping $\mathcal{M}_{AB} = \{\Phi\} : \mathcal{A} \to \mathcal{B}$ (see the Examples 30 and 31 in Section A.3.4 for more details).

In this way, for a given R-algebra α which satisfies the conditions for the mapping interpretations in Definition 66, we translate a logical representation of database map-

pings, based on SOtgds, into an algebraic representation based on relations of the instance databases and the functions obtained from mapping operads.

It is easy to verify that for a *query mapping* $\phi_{Ai}(\mathbf{x}) \Rightarrow r_B(\mathbf{t})$, a mapping interpretation α is an R-algebra such that the relation $\alpha(r_q)$ is just equal to the image of the function $\alpha(q_{A,i})$. The mapping interpretation of v_i is the transfer of information of this computed query into the relation $\alpha(r_B)$ of the database $\mathcal{B}$.

When α satisfies this query mapping $\phi_{Ai}(\mathbf{x}) \Rightarrow r_B(\mathbf{t})$, then $\alpha(r_q) \subseteq \alpha(r_B)$ and, consequently, the function $\alpha(v_i)$ is an injection, i. e., the *inclusion* of $\alpha(r_q)$ into $\alpha(r_B)$. Moreover, each R-algebra α of a given set of mapping operads between a source schema $\mathcal{A}$ and target schema $\mathcal{B}$ determines a particular information flux from the source into the target schema.

Definition 67 (Information flux). Let α be a mapping interpretation (an R-algebra in Definition 66) of a given set $\mathbf{M}_{AB} = \{q_1, \ldots, q_n, 1_{r_\emptyset}\} = \mathrm{MakeOperads}(\mathcal{M}_{AB})$ of mapping operads, obtained from an *atomic* mapping $\mathcal{M}_{AB} : \mathcal{A} \to \mathcal{B}$, and $A = \alpha^*(S_A)$ be an instance of the schema $\mathcal{A} = (S_A, \Sigma_A)$ that satisfies all constraints in Σ_A.

For each operation $q_i \in \mathbf{M}_{AB}$, $q_i = (e \Rightarrow (_)(\mathbf{t}_i)) \in O(r_{i,1}, \ldots, r_{i,k}, r'_i)$, let $\mathbf{x}_i$ be its tuple of variables, which appear at least one time free (not as an argument of a function) in $\mathbf{t}_i$ and appear as variables in the atoms of relational symbol of the schema $\mathcal{A}$ in the formula $e[(_)_j/r_{i,j}]_{1\leq j\leq k}$. Then we define:

(1) $\mathrm{Var}(\mathbf{M}_{AB}) = \bigcup_{1\leq i\leq n}\{\{x\} \mid x \in \mathbf{x}_i\}$.

We define the kernel of the information flux of $\mathbf{M}_{AB}$, for a given mapping-interpretation α, by (we denote the image of a function f by "im(f)")

(2) $\Delta(\alpha, \mathbf{M}_{AB}) = \{\pi_{\mathbf{x}_i}(\mathrm{im}(\alpha(q_i))) \mid q_i \in \mathbf{M}_{AB}, \text{ and } \mathbf{x}_i \text{ is not empty}\} \bigcup \perp^0$, if $\mathrm{Var}(\mathbf{M}_{AB}) \neq \emptyset$; $\perp^0$ otherwise.

We define the information flux from its kernel by

$$\mathrm{Flux}(\alpha, \mathbf{M}_{AB}) = T(\Delta(\alpha, \mathbf{M}_{AB})). \tag{A.20}$$

The flux of composition of $\mathbf{M}_{AB}$ and $\mathbf{M}_{BC}$ is defined by

$$\mathrm{Flux}(\alpha, \mathbf{M}_{BC} \circ \mathbf{M}_{AB}) = \mathrm{Flux}(\alpha, \mathbf{M}_{AB}) \bigcap \mathrm{Flux}(\alpha, \mathbf{M}_{BC}). \tag{A.21}$$

We say that an information flux is *empty* if it is equal to $\perp^0 = \{\perp\}$ (and hence it is not the empty set), analogously as for an empty instance database.

The information flux of the SOtgd of the mapping $\mathcal{M}_{AB}$ for the instance-level mapping $f = \alpha^*(\mathbf{M}_{AB}) : A \to \alpha^*(\mathcal{B})$ composed of the set of functions $f = \alpha^*(\mathbf{M}_{AB}) = \{\alpha(q_1), \ldots, \alpha(q_n), q_\perp\}$, is denoted by $\widetilde{f}$. Notice that $\perp \in \widetilde{f}$, and hence the information flux $\widetilde{f}$ is a instance database as well.

From this definition, each instance mapping is a set of functions whose information flux is the intersection of the information fluxes of all atomic instance mappings that compose this composed instance mapping. These basic properties of the instance mappings is used in order to define the database **DB** category where the instance mappings are the morphisms (i. e., the arrows) of this category, while the instance databases (each instance database is a set of relations of a schema also with the empty relation $\perp$) are its objects.

In the case of an atomic mapping, obtained from a set of egds of a given schema $\mathcal{A}$, we have the following particular property.

Corollary 29 ([5]). *Let* Φ *be the SOtgd equal to* $\text{EgdsToSOtgd}(\Sigma_A^{\text{egd}})$ *and* Ψ *be the SOtgd equal to* $\text{TgdsToConSOtgd}(\Sigma_A^{\text{tgd}})$ *for a given schema* $\mathcal{A} = (S_A, \Sigma_A^{\text{egd}} \bigcup \Sigma_A^{\text{tgd}})$.

Then, for every R-algebra α*, the information flux* $\text{Flux}(\alpha, \text{MakeOperads}(\{\Phi \wedge \Psi\})) = \perp^0$ *is empty.*

Note that this corollary confirms that, for any database schema $\mathcal{A} = (S_A, \Sigma_A)$ where $\Sigma_A = \Sigma_A^{\text{egd}} \bigcup \Sigma_A^{\text{tgd}}$, we can define the integrity-constraints mapping by a schema mapping $\top_{AA_\top} = \{\Phi \wedge \Psi\} : \mathcal{A} \to \mathcal{A}_\top$ (where Φ is the SOtgd equal to $\text{EgdsToSOtgd}(\Sigma_A^{\text{egd}})$ and Ψ is the SOtgd equal to $\text{TgdsToConSOtgd}(\Sigma_A^{\text{tgd}})$ and $\mathcal{A}_\top = (\{r_\top\}, \emptyset)$) and, consequently, by equivalent operads-mapping $\mathbf{T}_{AA_\top} = \text{MakeOperads}(\{\Phi\}) \bigcup \text{MakeOperads}(\{\Psi\}) : \mathcal{A} \to \mathcal{A}_\top$ with the *empty* information flux.

We have no mapping from $\mathcal{A}_\top$ into other schema mappings, so that this integrity-constraint mapping does not participate in any significant composition with other mappings in a given database mapping system. Moreover, from the fact that the information flux of composed mappings is equal to the intersection of the information fluxes of all atomic arrows which compose this mapping, such composed mapping which contains an atomic integrity-constraint mapping will always have an empty flux. Consequently, the role of the integrity-constraint mappings is only "logical," used to express the integrity constraints for schemas, and to verify if for a given mapping-interpretation α in Definition 66 they are satisfied. The extension of the database mapping systems with the integrity-constraints mappings will not modify its original semantic structure, but in this extended framework not only the interschema mappings but the schema integrity constraints will be the "first objects" of big data integration theory and presented in a uniform elegant manner.

Example 33. Schema $\mathcal{A}$ consists of a single binary relational symbol `Takes`, that associates student names with courses they take, i. e., $S_A = \{\texttt{Takes}(x_n, x_c)\}$, $\Sigma_A = \emptyset$. Schema $\mathcal{B}$ consists of one binary relational symbol `Takes1`, that is intended to provide a copy of `Takes`, and of an additional binary relational symbol `Student` that associates each student name with a student id, i. e., $S_B = \{\texttt{Takes1}(x_n, x_c), \texttt{Student}(x_n, y)\}$, $\Sigma_B = \emptyset$. Schema $\mathcal{D}$ consists of one binary relational symbol `Enrolment` that associates student ids with courses the student takes, and of an additional binary relational symbol `Teaching` that associates professor names with the courses, i. e., $S_D =$

$\{\texttt{Enrolment}(y, x_c), \texttt{Teaching}(x_p, x_c)\}$, $\Sigma_D = \emptyset$. Schema $\mathcal{C}$ consists of a one ternary relational symbol `Learning` that associates student name with courses he/she takes and professor names, and of an additional unary relational symbol `Professor` with professor names, i. e., $S_{\mathcal{C}} = \{\texttt{Learning}(x_n, x_c, x_p), \texttt{Professor}(x_p)\}$, and with integrity constraints

$$\begin{aligned}\Sigma_{\mathcal{C}} = \{&\forall x_p(\texttt{Professor}(x_p) \Rightarrow \exists x_n \exists x_c \texttt{Learning}(x_n, x_c, x_p)),\\ &\forall x_n \forall x_c \forall y \forall z((\texttt{Learning}(x_n, x_c, z_1) \wedge \texttt{Learning}(x_n, x_c, z_2)) \Rightarrow z_1 = z_2)\},\end{aligned}$$

where the second constraint defines the tuple (x_n, x_c) as a key of the relation `Learning`.

Let us consider the following composition of the atomic mappings:

$$\begin{aligned}&\mathcal{M}_{AB} : \mathcal{A} \to \mathcal{B}, \quad \mathcal{M}_{BD} : \mathcal{B} \to \mathcal{D}, \quad \mathcal{M}_{AC} : \mathcal{A} \to \mathcal{C} \quad \text{and} \quad \mathcal{M}_{CD} : \mathcal{C} \to \mathcal{D}, \quad \text{where}\\ &\mathcal{M}_{AB} = \{\forall x_n \forall x_c(\texttt{Takes}(x_n, x_c) \Rightarrow \texttt{Takes1}(x_n, x_c)),\\ &\qquad \forall x_n \forall x_c(\texttt{Takes}(x_n, x_c) \Rightarrow \texttt{Student}(x_n, f_1(x_n)))\},\\ &\mathcal{M}_{BD} = \{\forall x_n \forall x_c \forall y((\texttt{Takes1}(x_n, x_c) \wedge \texttt{Student}(x_n, y)) \Rightarrow \texttt{Enrolment}(y, x_c))\}\\ &\mathcal{M}_{AC} = \{\forall x_n \forall x_c(\texttt{Takes}(x_n, x_c) \Rightarrow \texttt{Learning}(x_n, x_c, f_2(x_n, x_c)))\},\\ &\mathcal{M}_{CD} = \{\forall x_n \forall x_c \forall x_p(\texttt{Learning}(x_n, x_c, x_p) \Rightarrow \texttt{Teaching}(x_p, x_c))\}.\end{aligned}$$

Hence, their composition is equal to the following mappings:

$$\begin{aligned}&\mathcal{M}_{AD} = \mathcal{M}_{BD} \circ \mathcal{M}_{AB} = \{\exists f_1(\forall x_1 \forall x_2(\texttt{Takes}(x_1, x_2) \Rightarrow \texttt{Enrolment}(f_1(x_1), x_2)))\},\\ &\mathcal{M}'_{AD} = \mathcal{M}_{CD} \circ \mathcal{M}_{AC} = \{\exists f_2(\forall x_1 \forall x_2(\texttt{Takes}(x_1, x_2) \Rightarrow \texttt{Teaching}(f_2(x_1, x_2), x_2)))\},\end{aligned}$$

which are not logically equivalent.

However, we obtain from Definition 67, for $A = \alpha^*(\mathcal{A})$, $B = \alpha^*(\mathcal{B})$ and $C = \alpha^*(\mathcal{C})$:

$$\begin{aligned}&\text{Flux}(\alpha, (\text{MakeOperads}(\mathcal{M}_{BD}) \circ \text{MakeOperads}(\mathcal{M}_{AB})))\\ &\quad = \text{Flux}(\alpha, \text{MakeOperads}(\mathcal{M}_{AB})) \bigcap \text{Flux}(\alpha, \text{MakeOperads}(\mathcal{M}_{BD}))\\ &\quad = T(\|\texttt{Takes}(x_n, x_c)\|_A, \pi_{x_n}\|\texttt{Takes}(x_n, x_c)\|_A)\\ &\qquad \bigcap T(\{\pi_{x_c, y}\|\texttt{Takes1}(x_n, x_c) \wedge \texttt{Student}(x_n, y)\|_B\}), \quad \text{and}\\ &\text{Flux}(\alpha, (\text{MakeOperads}(\mathcal{M}_{CD}) \circ \text{MakeOperads}(\mathcal{M}_{AC})))\\ &\quad = \text{Flux}(\alpha, \text{MakeOperads}(\mathcal{M}_{AC})) \bigcap \text{Flux}(\alpha, \text{MakeOperads}(\mathcal{M}_{CD}))\\ &\quad = T(\|\texttt{Takes}(x_n, x_c)\|_A) \bigcap T(\{\pi_{x_p, x_c}\|\texttt{Learning}(x_n, x_c, x_p)\|_C\}).\end{aligned}$$

In the case when in the given universe $\mathcal{U} = \mathbf{dom}$, the domain $\text{dom}(y)$ for the attribute y of the atom $\texttt{Student}(x_n, y)$ is disjoint from $\text{dom}(x_n)$ and from $\text{dom}(x_c)$ of the atom $\texttt{Takes}(x_n, x_c)$, and if $\text{dom}(x_p)$ for the attribute x_p of the atom $\texttt{Learning}(x_n, x_c, x_p)$ is disjoint from $\text{dom}(x_n)$ and from $\text{dom}(x_c)$ of the atom $\texttt{Takes}(x_n, x_c)$, then it is easy to verify that

$$\begin{aligned}&\mathrm{Flux}(\alpha, (\mathrm{MakeOperads}(\mathcal{M}_{BD}) \circ \mathrm{MakeOperads}(\mathcal{M}_{AB})))\\ &\quad = \mathrm{Flux}(\alpha, (\mathrm{MakeOperads}(\mathcal{M}_{CD}) \circ \mathrm{MakeOperads}(\mathcal{M}_{AC})))\\ &\quad = T(\|\mathtt{Takes}(x_n, x_c)\|_A).\end{aligned}$$

Consequently, in this case, the two composed mappings $\mathcal{M}_{AD}$ and $\mathcal{M}'_{AD}$ have the same information fluxes from $\mathcal{A}$ into target schema $\mathcal{D}$, so that, from the strict semantics point of view, they are equal instance-level mappings.

Now we can show that the information fluxes of the schema mappings have the categorial morphism's properties.

Corollary 30 ([5]). *The information fluxes of the schema mappings and the composition (i. e., the set intersection) of information fluxes satisfy categorial properties of the morphisms between instance databases as objects of such a category.*

Based on this and the corollary, it is possible to define the categorial semantics for the schema mappings, by defining a functor from the sketch category of a given schema-mapping graph into an instance-level category where the objects are the database instances and the morphisms are characterized by the information fluxes of the schema mappings.

Definition 68. Any two arrows $f, g : A \to B$ where A and B are the instance databases (the simple sets of the relations) in **DB** are *equal* if $\widetilde{f} = \widetilde{g}$, i. e., if they have the same information fluxes.

The formalization of the schema mappings by means of operads is useful in order to be able to extend each R-algebra α to a functor from the category sketch (obtained from a graph of (inter)schema mappings) into the base **DB** category, which represents the denotational semantics of this schema database mapping graph.

That is, each schema mapping $\mathcal{M}_{AB}$ transformed into its mapping-operad $\mathbf{M}_{AB} = \mathrm{MakeOperads}(\mathcal{M}_{AB}) = \{q_1, \ldots, q_n, 1_{r_0}\} : \mathcal{A} \to \mathcal{B}$ may be seen as a morphism of a sketch category, and hence it will be mapped by a functor (R-algebra) α, which satisfies Definition 66 (i. e., such that it is a *mapping interpretation*), into the **DB** category morphism $\alpha^*(\mathbf{M}_{AB}) = \{f_1, \ldots, f_n, q_\perp\} : A \to B$, where $A = \alpha(S_A)$ is an instance database of the schema $\mathcal{A}$, $B = \alpha(S_B)$ is an instance database of the schema $\mathcal{B}$ and $f_i = \alpha(q_i)$ are the functions obtained from the operads, with the domain equal to the Cartesian product of a subset of relations in the instance A and the codomain is a relation in the instance B.

The functors such that, for every operad's operation $q_i = v_i \cdot q_{A,i}$ in each sketch's mapping, the function $\alpha(v_i)$ is an inclusion (injection) will define a *model* of the schema database mapping system expressed by this sketch. This is the principal idea for the *categorical semantics* of the schema database mapping systems.

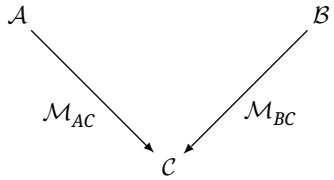
For the composition of *complex* database mapping graphs, it is important to distinguish the following two basic compositions of database schemas $\mathcal{A}$ and $\mathcal{B}$ with respect to DBMSs:

– In the composed schema, in order to make it impossible to write a query over the composition with relations of both databases, the two database schemas should be mutually *separated* by two independent DBMSs: it is a common case when two databases are separated, and this binary separation-composition at schema level is denoted by $\mathcal{A}\dagger\mathcal{B}$ (disjoint union) where all elements are indexed by schema symbols (denoted by a label $\ddot{A}$, for a schema $\mathcal{A} = (S_A, \Sigma_A)$), equal to the complex schema $(S_{A\dagger B}, \Sigma_{A\dagger B})$ where $S_{A\dagger B} = (\{\ddot{A}\} \times S_A) \bigcup (\{\ddot{B}\} \times S_B)$ and $\Sigma_{A\dagger B} = (\{\ddot{A}\} \times \Sigma_A) \bigcup (\{\ddot{B}\} \times \Sigma_B)$ and $\times$ is Cartesian product of sets (if $\mathcal{B} = \mathcal{A}$, then we set $\ddot{A} = 1$ and $\ddot{B} = 2$).
– In the composed schema, when the two database schemas are *connected* into the same DBMS (without any change of the two original database schemas): in this case, we are able to use the queries over this composed schema with relations *of both* databases for interdatabase mappings. This binary federation-composition, denoted by $\mathcal{A} \bigoplus \mathcal{B}$, is the (simple) *union* but with *renaming* (if we have the same relational symbol $r \in S_A \bigcap S_B$, then before the union, we rename this symbol in S_B by a new fresh relational symbol of the same type), so that $S_{\mathcal{A} \bigoplus \mathcal{B}} = S_A \overset{\bigcup}{\underset{\mathrm{rn}}{}} S_B$, where $\overset{\bigcup}{\underset{\mathrm{rn}}{}}$ denotes the union with renaming.

The equality $'='$ for database schemas is naturally defined as follows: for any two schemas $\mathcal{A}, \mathcal{B} \in \mathbb{S}$, $\mathcal{A} = \mathcal{B}$ if $\pi_i(\mathcal{A}) = \pi_i(\mathcal{B})$, $i = 1, 2$. Let us consider the mappings $\mathcal{M} : \mathcal{A}\dagger\mathcal{B} \to \mathcal{C}$ and $\mathcal{M} : \mathcal{A} \bigoplus \mathcal{B} \to \mathcal{C}$. In the first case, in any query mapping $q(\mathbf{x}) \Rightarrow q_C(\mathbf{x})$ in the SOtgd of $\mathcal{M}$, all relational symbols in the query $q(\mathbf{x})$ must be of the database $\mathcal{A}$ or (mutually exclusive) of the database $\mathcal{B}$, i. e.,

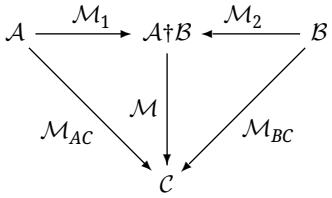
(1) $\mathcal{M} = \{\exists \mathbf{f}(\forall \mathbf{x}_1(q_{A,1}(\mathbf{x}_1) \Rightarrow q_{C,1}(\mathbf{t}_1)) \wedge \cdots \wedge \forall \mathbf{x}_k(q_{A,k}(\mathbf{x}_k) \Rightarrow q_{C,k}(\mathbf{t}_k)) \wedge \forall \mathbf{y}_1(q_{B,1}(\mathbf{y}_1) \Rightarrow q_{C,k+1}(\mathbf{t}'_1)) \wedge \cdots \wedge \forall \mathbf{y}_m(q_{B,m}(\mathbf{y}_m) \Rightarrow q_{C,k+m}(\mathbf{t}'_m)))\}$,

with $\mathbf{t}_i$ the tuple of terms with variables in $\mathbf{x}_i$, for $1 \leq i \leq k$, and $\mathbf{t}'_i$ the tuple of terms with variables in $\mathbf{y}_i$, for $1 \leq i \leq m$. Consequently, this mapping is the graph:



with $\mathcal{M}_{AC} = \{\exists \mathbf{f}_A(\forall \mathbf{x}_1(q_{A,1}(\mathbf{x}_1) \Rightarrow q_{C,1}(\mathbf{t}_1)) \wedge \cdots \wedge \forall \mathbf{x}_k(q_{A,k}(\mathbf{x}_k) \Rightarrow q_{C,k}(\mathbf{t}_k)))\}$ and $\mathcal{M}_{BC} = \{\exists \mathbf{f}_B(\forall \mathbf{y}_1(q_{B,1}(\mathbf{y}_1) \Rightarrow q_{C,k+1}(\mathbf{t}'_1)) \wedge \cdots \wedge \forall \mathbf{y}_m(q_{B,m}(\mathbf{y}_m) \Rightarrow q_{C,k+m}(\mathbf{t}'_m)))\}$, and $\mathbf{f}_A, \mathbf{f}_B \subseteq \mathbf{f}$, while in the case of mapping $\mathcal{M} : \mathcal{A} \bigoplus \mathcal{B} \to \mathcal{C}$ such a decomposition is not possible because we can have a query mapping $q(\mathbf{x}) \Rightarrow q_C(\mathbf{x})$ in the SOtgd of $\mathcal{M}$ with relational symbols in $q(\mathbf{x})$ from both databases $\mathcal{A}$ and $\mathcal{B}$.

If we introduce the mappings $\mathcal{M}_1 = \{\bigwedge\{\forall \mathbf{x}_i(r_{Ai}(\mathbf{x}_i) \Rightarrow r_{Ai}(\mathbf{x}_i)) \mid r_{Ai} \in \mathcal{A}\}\}$ and $\mathcal{M}_2 = \{\bigwedge\{\forall \mathbf{y}_i(r_{Bi}(\mathbf{y}_i) \Rightarrow r_{Bi}(\mathbf{y}_i)) \mid r_{Bi} \in \mathcal{B}\}\}$, then we obtain the mapping graph

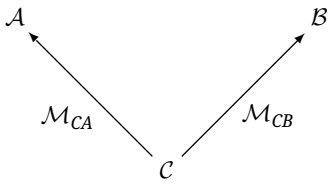


that can be seen as a cocone diagram for schema database mappings.

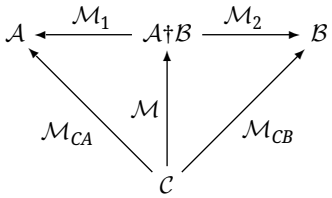
Let us consider another dual example, a mapping $\mathcal{M} : \mathcal{C} \to \mathcal{A}\dagger\mathcal{B}$. In this case, in any query mapping $q_C(\mathbf{x}) \Rightarrow q(\mathbf{x}) \in \mathcal{M}$, all relational symbols in the query $q(\mathbf{x})$ must be of database $\mathcal{A}$ or (mutually exclusive) of database $\mathcal{B}$. That is,

(2) $\mathcal{M} = \{\exists \mathbf{f}(\forall \mathbf{x}_1(q_{C,1}(\mathbf{x}_1) \Rightarrow q_{A,1}(\mathbf{t}_1)) \wedge \cdots \wedge \forall \mathbf{x}_k(q_{C,k}(\mathbf{x}_k) \Rightarrow q_{A,k}(\mathbf{t}_k)) \wedge \forall \mathbf{y}_1(q_{C,k+1}(\mathbf{y}_1) \Rightarrow q_{B,1}(\mathbf{t}'_1)) \wedge \cdots \wedge \forall \mathbf{y}_m(q_{C,k+m}(\mathbf{y}_m) \Rightarrow q_{B,m}(\mathbf{t}'_m)))\}$.

Consequently, this mapping can be equivalently represented by the graph:



where $\mathcal{M}_{CA} = \{\exists \mathbf{f}_A(\forall \mathbf{x}_1(q_{C,1}(\mathbf{x}_1) \Rightarrow q_{A,1}(\mathbf{t}_1)) \wedge \cdots \wedge \forall \mathbf{x}_k(q_{C,k}(\mathbf{x}_k) \Rightarrow q_{A,k}(\mathbf{t}_k)))\}$ and $\mathcal{M}_{BC} = \{\exists \mathbf{f}_B(\forall \mathbf{y}_1(q_{B,1}(\mathbf{y}_1) \Rightarrow q_{C,k+1}(\mathbf{t}'_1)) \wedge \cdots \wedge \forall \mathbf{y}_m(q_{B,m}(\mathbf{y}_m) \Rightarrow q_{C,k+m}(\mathbf{t}'_m)))\}$ If we again introduce the mappings $\mathcal{M}_1 = \{\bigwedge\{\forall \mathbf{x}_i(r_{Ai}(\mathbf{x}_i) \Rightarrow r_{Ai}(\mathbf{x}_i)) \mid r_{Ai} \in \mathcal{A}\}\}$ and $\mathcal{M}_2 = \{\bigwedge\{\forall \mathbf{y}_i(r_{Bi}(\mathbf{y}_i) \Rightarrow r_{Bi}(\mathbf{y}_i)) \mid r_{Bi} \in \mathcal{B}\}\}$, then



is a dual graph that can be seen as a cone diagram for the schema database mappings.

It is important to notice that differently from the mapping graphs, where the schema mappings are expressed by SOtgds, the small sketch categories derived from such a graph will have the equivalent schema mappings but represented algebraically by the sets of operad's operations, in order to be able to represent each R-algebra α as a *functor* from this sketch category (at the schema level) into the denotational (instance level) **DB** category. For the mappings that involve the composed source schema (by separation operator †) $\mathcal{M} = \{\Phi\} : \mathcal{A}\dagger\mathcal{B} \to \mathcal{C}$, we cannot directly use the algorithm MakeOperads($\mathcal{M}$)

because we would obtain operad's operations possibly with relations of both schemas. Consequently, we have to divide this SOtgd Φ into a conjunction $\Phi_A \wedge \Phi_B$, where Φ_A is an SOtgd whose left-hand sides of implication contain only relational symbols in $\mathcal{A}$ and Φ_B, only relational symbols in $\mathcal{B}$ (as in the mapping (1)).

Consequently, we define the complex mapping-operad $\mathbf{M}_{(A\dagger B)C}$ by

(3) $\mathbf{M}_{(A\dagger B)C} \triangleq [\mathbf{M}_{AC}, \mathbf{M}_{BC}] : \mathcal{A}\dagger\mathcal{B} \to \mathcal{C}$

where $\mathbf{M}_{AC} \triangleq \text{MakeOperads}(\{\Phi_A\}) = (q_1, \ldots, q_k, 1_{r_\emptyset}) : \mathcal{A} \to \mathcal{C}$ and $\mathbf{M}_{BC} \triangleq \text{MakeOperads}(\{\Phi_B\} = (q'_1, \ldots, q'_m, 1_{r_\emptyset}) : \mathcal{B} \to \mathcal{C}$ are two simple mapping operads by introducing a structural-operator $[_, _]$. Information flux of complex mapping operads is different from the information flux of ordinary (simple) mapping operads:

(4) $\text{Flux}(\alpha, [\mathbf{M}_{AC}, \mathbf{M}_{BC}]) \triangleq \text{Flux}(\alpha, \mathbf{M}_{AC}) \biguplus \text{Flux}(\alpha, \mathbf{M}_{BC}) = \{(1, a) \mid a \in \text{Flux}(\alpha, \mathbf{M}_{AC})\} \bigcup \{(2, b) \mid b \in \text{Flux}(\alpha, \mathbf{M}_{BC})\}$.

Analogously, each complex mapping-operad $\mathbf{M}_{D(A\dagger B)}$ from a simple (nonseparated) schema $\mathcal{D}$ into separated schema $\mathcal{A}\dagger\mathcal{B}$ is denoted as $\langle \mathbf{M}_{DA}, \mathbf{M}_{\text{DB}} \rangle : \mathcal{D} \to \mathcal{A}\dagger\mathcal{B}$ where $\mathbf{M}_{DA} : \mathcal{D} \to \mathcal{A}$ and $\mathbf{M}_{\text{DB}} : \mathcal{D} \to \mathcal{B}$ are ordinary (simple) mapping-operads (equal to $\{1_{r_\emptyset}\}$ if they are not defined), with

(5) $\text{Flux}(\alpha, \langle \mathbf{M}_{DA}, \mathbf{M}_{\text{DB}} \rangle) \triangleq \text{Flux}(\alpha, \mathbf{M}_{DA}) \biguplus \text{Flux}(\alpha, \mathbf{M}_{\text{DB}})$.

Any pair of simple mapping-operads $\mathbf{M}_{AC}, \mathbf{M}_{BD}$ can be expressed by single complex mapping operad:

(6) $\mathbf{M}_{(A\dagger B)(C\dagger D)} = \mathbf{M}_{AC}\dagger\mathbf{M}_{BD} \triangleq \mathbf{M}_{AC} \biguplus \mathbf{M}_{BD} : \mathcal{A}\dagger\mathcal{B} \to \mathcal{C}\dagger\mathcal{D}$, with $\text{Flux}(\alpha, \mathbf{M}_{AC} \biguplus \mathbf{M}_{BD}) \triangleq \text{Flux}(\alpha, \mathbf{M}_{DA}) \biguplus \text{Flux}(\alpha, \mathbf{M}_{\text{DB}})$.

Now we introduce formally these types of complex mapping operads as follows.

Definition 69. The n-ary composition of separated simple schemas define the following four types of complex morphisms:

1. A mapping-operad $\biguplus(\mathbf{M}_{A_1C_1}, \ldots, \mathbf{M}_{A_nC_n}) : \mathcal{A}_1\dagger\cdots\dagger\mathcal{A}_n \to \mathcal{C}_1\dagger\cdots\dagger\mathcal{C}_n$, with

$$\text{Flux}\Big(\alpha, \biguplus(\mathbf{M}_{A_1C_1}, \ldots, \mathbf{M}_{A_nC_n})\Big) = \biguplus(\text{Flux}(\alpha, \mathbf{M}_{A_1C_1}), \ldots, \text{Flux}(\alpha, \mathbf{M}_{A_nC_n})).$$

2. A mapping-operad $[\mathbf{M}_{A_1C}, \ldots, \mathbf{M}_{A_nC}] : \mathcal{A}_1\dagger\cdots\dagger\mathcal{A}_n \to \mathcal{C}$, with

$$\text{Flux}(\alpha, [\mathbf{M}_{A_1C}, \ldots, \mathbf{M}_{A_nC}]) = \biguplus(\text{Flux}(\alpha, \mathbf{M}_{A_1C}), \ldots, \text{Flux}(\alpha, \mathbf{M}_{A_nC})).$$

3. A mapping-operad $\langle \mathbf{M}_{DA_1}, \ldots, \mathbf{M}_{DA_n} \rangle : D \to \mathcal{A}_1\dagger\cdots\dagger\mathcal{A}_n$, with

$$\text{Flux}(\alpha, \langle \mathbf{M}_{DA_1}, \ldots, \mathbf{M}_{DA_n} \rangle) = \biguplus(\text{Flux}(\alpha, \mathbf{M}_{DA_1}), \ldots, \text{Flux}(\alpha, \mathbf{M}_{DA_n})).$$

4. The composition of last two types of complex morphisms is a mapping operad from $\mathcal{D}$ into $\mathcal{C}$:

$$\begin{aligned}\mathbf{M}_{DC} &= [\mathbf{M}_{A_1C},\dots,\mathbf{M}_{A_nC}]\circ\langle M_{DA_1},\dots,\mathbf{M}_{DA_n}\rangle\\ &= \lceil \mathbf{M}_{A_1C}\circ\mathbf{M}_{DA_1},\dots,\mathbf{M}_{A_nC}\circ\mathbf{M}_{DA_n}\rfloor : \mathcal{D}\to\mathcal{A}_1\dagger\cdots\dagger\mathcal{A}_n\to\mathcal{C},\end{aligned}$$

with a set of nonempty mappings $\mathbf{M}_{A_iC}\circ\mathbf{M}_{DA_i} : \mathcal{D}\to\mathcal{C}$, $i = 1,\dots,n$, enclosed by structural operator $\lceil_,\dots,_\rfloor$, all with the same source and target schemas, so that the resulting information flux is a flux of the union of all these mappings:

$$\mathrm{Flux}(\alpha,\mathbf{M}_{DC}) = T\Big(\bigcup\{\mathrm{Flux}(\alpha,\mathbf{M}_{A_iC}\circ\mathbf{M}_{DA_i}) \mid 1\le i\le n\}\Big).$$

5. Each schema mapping above, with structural operators $\biguplus$, $[,]$, $\langle,\rangle$ and $\lceil,\rfloor$, is denominated as a *complex* $\biguplus$-atomic sketch's schema mapping.

For a given mapping-interpretation α, we obtain the instance-database complex morphisms, e. g., $\alpha^*([\mathbf{M}_{A_1C},\dots,\mathbf{M}_{A_nC}]) = [\alpha^*(\mathbf{M}_{A_1C}),\dots,\alpha(\mathbf{M}_{A_nC})]$. We also introduce another kind of binary symmetric operations for a composition of the schemas and schema mappings, denoted by $\bigcap_\alpha$, where α is an R-algebra.

Definition 70. For each R-algebra α, we define an α-intersection symmetric binary operator $\bigcap_\alpha$ for the schema mappings, as follows:

1. For any two simple schemas $\mathcal{A}$ and $\mathcal{B}$, their α-intersection is defined by

$$\mathcal{A}\bigcap_\alpha\mathcal{B} = (S,\emptyset),\qquad \text{where } S = \Big\{r\in\mathbb{R} \mid \alpha(r)\in T(\alpha^*(\mathcal{A}))\bigcap T(\alpha^*(\mathcal{B}))\Big\}.$$

2. For any two simple schema mappings $\mathcal{M}_{AC} : \mathcal{A}\to\mathcal{C}$ and $\mathcal{M}_{BD} : \mathcal{B}\to\mathcal{D}$, the α-intersection of these two mappings is defined by

$$\begin{aligned}\mathcal{M}_{AC}\bigcap_\alpha\mathcal{M}_{BD} = \{\Phi\} : \mathcal{A}\bigcap_\alpha\mathcal{B}\to\mathcal{C}\bigcap_\alpha\mathcal{D},\quad &\text{where } \Phi \text{ is a tgd}\\ \bigwedge\Big\{\forall\mathbf{x}(r(\mathbf{x})\Rightarrow r(\mathbf{x})) \mid r\in\mathcal{A}\bigcap_\alpha\mathcal{B} &\text{ and } \alpha(r)\in\mathrm{Flux}(\alpha,\mathrm{MakeOperads}(\mathcal{M}_{AC}))\\ \bigcap\mathrm{Flux}(\alpha,\mathrm{MakeOperads}(\mathcal{M}_{BD}))\Big\}.&\end{aligned}$$

It is easy to show that for a simple schema $\mathcal{A}$ and R-algebra α, with the instance-database $A = \alpha^*(\mathcal{A})$, the schema of the instance-database TA is equal to $\mathcal{A}\bigcap_\alpha\mathcal{A}$, i. e., $TA = \alpha^*(\mathcal{A}\bigcap_\alpha\mathcal{A})$.

Moreover, for $A = \alpha^*(\mathcal{A})$ and $B = \alpha^*(\mathcal{B})$, $TA\bigcap TB = \alpha^*(\mathcal{A}\bigcap_\alpha\mathcal{B})$.

Analogously, for the α-intersection of schema mappings, it holds that

$$\mathrm{Flux}\Big(\alpha,\mathcal{M}_{AC}\bigcap_\alpha\mathcal{M}_{BD}\Big) = \mathrm{Flux}(\alpha,\mathrm{MakeOperads}(\mathcal{M}_{AC}))\bigcap\mathrm{Flux}(\alpha,\mathrm{MakeOperads}(\mathcal{M}_{BD})).$$

The schema separation, connection and α-intersection are necessary in order to manage complex database mappings. The category-theoretic data model that we use has come to be known as the *sketch data model*.

Notice that, different from the work dedicated to categorical semantics of Entity-Relationship internal relational database models where nodes of sketches are single relations, here, at a higher level of abstraction, the nodes are whole databases. Consequently, in such a framework, we do not use commutative database mapping systems, and hence D is an empty set. In fact, in a database mapping system, the (co)cone diagrams above will never be used in practical representations of database mapping systems. Instead of that, they will be alternatively used only for their self-consistent parts, as a first diagram above, or, equivalently, as a single arrow $\mathcal{M} : \mathcal{A}\dagger\mathcal{B} \to \mathcal{C}$. However, for the introduced schema composition operator †, the above cone and cocone diagrams have to be presented in $\mathcal{C}$ for our sketches.

A.4 Introduction to field theory and symmetries

In the classical Field Theory (FT) framework, the vacuum corresponds to the zero value, and each wave-packet Ψ can be considered a complex field, i. e., a mapping from the Minkowski time-space manifold with coordinates $(t, \vec{\mathbf{r}}) = (t, x, y, z)$ into the field of complex numbers. In theoretical physics, Minkowski space is often contrasted with Euclidean space. While a Euclidean space has only spacelike dimensions $q_0 = t$, a Minkowski space also has one timelike dimension. In the pictorial representations, the time axis is represented as vertical and upward-oriented, where the ascending arrow indicates the future. The Euclidean space $\mathbb{R}^3$ with $t = 0$ is then considered as horizontal.

All the world lines of light rays emitted from the event O by any (moving or at rest) light emitter are represented in a four-dimensional real vector space $M^4 = \mathbb{R}^{1,3}$ by the linear generatrices of the cone with $\|\vec{\mathbf{r}}\| = q_0 \equiv ct$, $t > 0$, which is called the "future light cone of O." The whole set of light world lines passing at O is the set of generatrices of the "light cone C of O," with quadratic equation $\|\mathbf{r}_4\|^2 = \mathbf{r}_4^2 \equiv (ct)^2 - \|\vec{\mathbf{r}}\|^2 = (ct)^2 - (x^2 + y^2 + z^2) = 0$. All trajectories inside this light cone are with velocities v less then c, so that for them the distance $ds^2 = (cd\tau)^2 = (cdt)^2 - (vdt)^2 = (cdt)^2 - (dx^2 + dy^2 + dz^2) \geq 0$ as required by the positive metrics with $ds^2 = (cdt)^2 + (idx^2) + (idy)^2 + (idz)^2$, where t, x, y, z are any system of inertial coordinates, and hence we will chose the real time dimension and imaginary three Euclidean dimensions as in [168] (with metric signature $(+, -, -, -)$).

Consequently, a vector of position in this time-space 4-dimensional system is given by the *Minkowski tetrad* (4-dimensional vector base) $(\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ and the reals $\eta_{ij} = \mathbf{e}_i \cdot \mathbf{e}_j$, where the matrix (η_{ij}) of the metric tensor (and its inverse (η^{ij})) is given by

$$(\eta_{ij}) = (\eta^{ij}) = \mathrm{diag}(1,-1,-1,-1) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{A.22}$$

so that any point of this time space is given by a *position four-dimensional vector* (or *Minkowski tetrad* [168]), $\mathbf{r}_4 = q_0\mathbf{e}_0 + q_1\mathbf{e}_1 + q_2\mathbf{e}_2 + q_3\mathbf{e}_3 = ct\mathbf{e}_0 + x\mathbf{e}_1 + y\mathbf{e}_2 + z\mathbf{e}_3 = ct\mathbf{e}_0 + \vec{\mathbf{r}}$, where t is the time (i. e., ct is the timelike component of $\vec{\mathbf{r}}_4$, where c is the velocity of light in the vacuum) and $\vec{\mathbf{r}} = x\mathbf{e}_1 + y\mathbf{e}_2 + z\mathbf{e}_3$ is an ordinary Euclidean vector with the three spatial coordinates x, y, z. Its infinitesimal amount is defined by $d\mathbf{s} = cdt\mathbf{e}_0 + dx\mathbf{e}_1 + dy\mathbf{e}_2 + dz\mathbf{e}_3$, where dt, dx, dy and dz are infinitesimal amounts of time-space dimensions. From the fact that the space-dimensions unit vectors are imaginary, the conjugate vector is equal to $\mathbf{r}_4^* = ct\mathbf{e}_0 - \vec{\mathbf{r}}$.

Then the inner (scalar) product $\langle \mathbf{r}_4, \mathbf{r}_4' \rangle = (\mathbf{r}_4)^T(\eta_{ij})(\mathbf{r}_4')$ (where $(\mathbf{r}_4) = (ct, x, y, z)$ is a tetrad matrix form of the four-dimensional vector $\mathbf{r}_4$) is equal to

$$\mathbf{r}_4 \cdot \mathbf{r}_4' = (q_i\mathbf{e}_i) \cdot (q_j'\mathbf{e}_j) = \sum_{i,j} q_i x_j' (\mathbf{e}_i \cdot \mathbf{e}_j) = \sum_{i,j} q_i q_j' \eta_{ij} = q_0 q_0' - q_1 q_1' - q_2 q_2' - q_3 q_3',$$

written simply $\mathbf{r}_4\mathbf{r}_4'$ as well. A particular case of scalar product is the *Lorentz or Minkowski norm* $\|\mathbf{r}_4\| = \sqrt{\mathbf{r}_4\mathbf{r}_4}$, so that the scalar product can be defined in terms of this Lorentz (or Minkowski) norm by $\mathbf{r}_4 \cdot \mathbf{r}_4' = \frac{1}{2}\|\mathbf{r}_4 + \mathbf{r}_4'\|^2 - \|\mathbf{r}_4\|^2 - \|\mathbf{r}_4'\|^2$. The 4-vector $\mathbf{r}_4$ is called:

1. *timelike* if $\|\mathbf{r}_4\|^2 > 0$;
2. *spacelike* if $\|\mathbf{r}_4\|^2 < 0$;
3. *null* if $\|\mathbf{r}_4\|^2 = 0$.

It is *causal* if it is timelike or null. Minkowski vector space can be regarded as the space of position vectors, relative to an arbitrarily chosen origin O, of the points (events), which constitute Minkowski time space. For any pair of points $\mathbf{r}_4$ and $\mathbf{r}_4'$ we define the *squared interval* $\Delta(\mathbf{r}_4, \mathbf{r}_4) = \|\mathbf{r}_4 - \mathbf{r}_4'\|$, so that for the chosen origin O defined by the point $\mathbf{r}_4'$, the light cone of $\mathbf{r}_4'$ is defined by the set of all points $\mathbf{r}_4$ such that $\Delta(\mathbf{r}_4, \mathbf{r}_4') = 0$. Thus, in this 4-dimensional system (Minkowski tetrad) the time is real while the three orthogonal space coordinates are imaginary. This choice is adopted in order to have that the distance, $d\mathbf{s}^2 = d\mathbf{s}d\mathbf{s} = (cdt)^2 - dx^2 - dy^2 - dz^2$, for all local time-space reference systems of observations of quantum events be the positive real value (where space dimensions are limited).

Galileo's classical principle of inertial relativity asserts that for any material object in any state of motion there exists a system of space and time coordinates, called *inertial coordinates*, w. r. t. which the object is instantaneously at rest and inertia is homogeneous and isotropic (the latter being necessary for Newton's laws of motion to hold at least quasi-statically: for the motion of material objects the most important symmetries are the isotropy of inertia, the conservation of momentum, the law of equal action and

reaction, and so on). "Isotropic" space means that the inertia of an object does not depend on any absolute reference direction in space: it is sufficient to say the resistance to acceleration of a resting object is the same in all 3-D spatial directions. In fact, Newton"s "laws of motion" assert the inertia is homogeneous and isotropic w. r. t. inertial coordinate systems.

> "Every body continues its state of rest, or of uniform motion in a right line, unless it is compelled to change that state by the forces impressed upon it."

What is important in the 4-dimensional Minkowski space M^4, with its points $(t, \vec{\mathbf{r}}) \in M^4$, that it *is absolute* and provides rigorous criteria for inertial and accelerated motion. A free particle, which moves by inertia (constant velocity), is a straight world line in Minkowski time space, whereas the timelike world line of an accelerating particle is clearly different—it is *curved* (i. e., deformed). These criteria show that in time space the absoluteness of inertial (nonresistant) and accelerated (resistant) motion become more understandable. These absolute properties of world line (straightness and deformation) correspond to the absoluteness (frame independence) of inertial and accelerated motion—it is an experimental fact that a particle moving by inertia offers no resistance to its uniform motion, while an accelerating particle resists its acceleration. A body moving by inertia is represented by geodesic world line. Absolute acceleration is a mere manifestation of the deformation of the world line of an accelerating particle and *does not imply some absolute space* with respect to which the particle accelerates. From these Minkowski results, we are led to the surprising insight about the origin of inertia—the resistance a particle offers to its acceleration is in fact the static resistance in the deformed world line of the accelerating particle. The world line of a single particle in the universe is either geodesic or deformed, which means that the particle is either moving by inertia or accelerating. And this is basis of general relativity when Einstein linked gravitation with the geometry of time space, where the gravity is not a force but a manifestation of the curvature of time space. In general, relativity a particle, whose world line is geodesic, is a free particle, which moves by inertia. Thanks to Einstein, we know that electromagnetism is fundamentally different from gravitation (at least in its original General Relativity (GR)): electromagnetic phenomena are caused by photons (electromagnetic forces), whereas gravitational phenomena are the manifestation of the non-Euclidean geometry of time space (different from the photons, the existence of gravitons is not demonstrated).

The starting point in the field theory is the Lagrangian density $\mathcal{L}$, from which, by using the Euler–Lagrange equation we can obtain the field partial differential equation (by minimizing the action $\int d^4q\mathcal{L}$). In effect, the principle of least action can be extended to field theory, where rather than the position of a particle, the values of a field are considered. The history between two fixed field configurations, for which the action is an extremum will be the realized history. The theory is *deterministic* if this history is unique, i. e., if there is only one history that extremizes the action between two fixed

configurations. We will show in this monograph that the history of the particle's new field (wave-packet) Ψ is unique, so that this completion theory of QM is deterministic. Interactions between particles are due to terms in the Lagrangian, which involves the fields of every considered particle.

In classical physics, the primary reason for introducing the concept of the field is to construct laws of nature that are local. In quantum mechanics, the role of the fields are necessary when we consider the fact that the particles can be created and destroyed as well, so that the number of particles continuously change. Such events happen in the interactions of particle–antiparticle pairs, when a particle of mass m_0 is localized within a distance of order $\lambda_c = \frac{\hbar}{2m_0}$, called the Compton wavelength (it is always smaller than the de Broglie wavelength $\lambda = \frac{2\pi\hbar}{p}$, i. e., the distance at which the wavelike nature of pointlike particles (used in standard quantum theory) becomes apparent).

In what follows, in order to simplify the mathematical formulas, we will introduce the indexed coordinates (q_0, q_1, q_2, q_3) with $q_0 = ct$, instead of the standard time-space coordinates $(t, \vec{\mathbf{r}}) = (t, x, y, z)$, the variables of the velocity components v_1, v_2, v_3 and ∂_j instead of $\frac{\partial}{\partial q_j}$ for $j = 0, 1, 2, 3$.

Thus, the Lagrangian density $\mathcal{L}$ is a function of independent variables q_j and dependent variables v_j, Ψ and their derivatives up to a fixed but arbitrary order, $\partial_j\Psi$. Thus, for any closed finite time-interval $[t_0, t_1]$ and time-space volume $\Omega_4 = [t_0, t_1] \times \mathbb{R}^3$, the action of a field theory with Lagrangian density $\mathcal{L}$ is defined by

$$S[\Psi] = \int_{t_0}^{t_1} dtL = \int_{\Omega_4} d^4q\mathcal{L}(\Psi, \partial_j\Psi, q_j, v_j). \tag{A.23}$$

The action S is a functional of the field Ψ and we are looking for a field configuration Ψ that extremizes the action. The field is written as $\Psi' = \Psi + \delta\Psi$ where Ψ is the field configuration (i. e., matter density) that extremizes the action and $\delta\Psi$ is a variation, which vanishes on the boundary $\triangle\Omega_4$. The action becomes

$$S[\Psi + \delta\Psi] = \int_{\Omega_4} d^4q\mathcal{L}\big(\Psi + \delta\Psi, \partial_j(\Psi + \delta\Psi), q_j, v_j\big). \tag{A.24}$$

So that, by expanding the $\mathcal{L}$ to first order in $\delta\Psi$, we obtain

$$\begin{aligned}\mathcal{L}\big(\Psi + \delta\Psi, \partial_j(\Psi + \delta\Psi), q_j, v_j\big) &= \mathcal{L}(\Psi, \partial_j\Psi, q_j, v_j) + \delta\mathcal{L}\\ \delta\mathcal{L} &= \frac{\partial\mathcal{L}}{\partial\Psi}\delta\Psi + \sum_{j=0}^{3}\frac{\partial\mathcal{L}}{\partial(\partial_j\Psi)}\partial_j(\delta\Psi).\end{aligned} \tag{A.25}$$

From equation (A.24), the action S is expanded to first order in $\delta\Psi$, as follows:

$$S[\Psi+\delta\Psi] = \int\limits_{\Omega_4} d^4q\mathcal{L}(\Psi,\partial_j\Psi,q_j,v_j) + \int\limits_{\Omega_4} d^4q\left[\frac{\partial\mathcal{L}}{\partial\Psi}\delta\Psi + \sum_{j=0}^{3}\frac{\partial\mathcal{L}}{\partial(\partial_j\Psi)}\delta(\partial_j\Psi)\right]$$
$$= S[\Psi] + \int\limits_{\Omega_4} d^4q\left[\frac{\partial\mathcal{L}}{\partial\Psi}\delta\Psi + \sum_{j=0}^{3}\frac{\partial\mathcal{L}}{\partial(\partial_j\Psi)}\partial_j(\delta\Psi)\right],$$

where the second term is equal to δS, so

$$\delta S = \int\limits_{\Omega_4} d^4q\left[\frac{\partial\mathcal{L}}{\partial\Psi}\delta\Psi + \sum_{j=0}^{3}\frac{\partial\mathcal{L}}{\partial(\partial_j\Psi)}\partial_j(\delta\Psi)\right]$$
$$= \int\limits_{\Omega_4} d^4q\left[\frac{\partial\mathcal{L}}{\partial\Psi} - \sum_{j=}^{3}\partial_j\frac{\partial\mathcal{L}}{\partial(\partial_j\Psi)}\right] + \delta\Psi\left[\frac{\partial\mathcal{L}}{\partial(d_0\Psi)} + \sum_{j=0}^{3}\frac{\partial\mathcal{L}}{\partial(\partial_j\Psi)}\right]\Bigg|_{\triangle\Omega_4}$$
$$= \int\limits_{\Omega_4} d^4q\left[\frac{\partial\mathcal{L}}{\partial\Psi} - \sum_{j=0}^{3}\partial_j\frac{\partial\mathcal{L}}{\partial(\partial_j\Psi)}\right]\delta\Psi.$$

Thus, from the variation principle we have that $\delta S = 0$ and since the variation $\delta\Psi$ is arbitrary, the term in the parentheses of the integral above must vanish, i. e., we obtain the following Euler–Lagrange equation for the field Ψ:

$$\frac{\partial\mathcal{L}}{\partial\Psi} = \sum_{j=0}^{3}\partial_j\left(\frac{\partial\mathcal{L}}{\partial(\partial_j\Psi)}\right). \tag{A.26}$$

Symmetries play a fundamental role in physics because they are related to conservation laws. This is stated in Noether's theorem, which says that invariance of the action under a symmetry transformation implies the existence of a conserved quantity. For instance, the conservation of the momentum vector $\vec{\mathbf{p}}$ is associated with translation invariance of the Lagrangian, i. e., the Poincare transformation $\vec{\mathbf{r}} \to \vec{\mathbf{r}} + \vec{\mathbf{r}_0}$, where $\vec{\mathbf{r}_0}$ is a constant vector; while the conservation of energy comes from the invariance under time translations $t \to t + t_0$. Assume the Lagrangian density $\mathcal{L}$ be invariant under a transformation of the matter field Ψ, so that for an infinitesimal transformation $\Psi' = \Psi + \delta\Psi$, we obtain by considering the Euler–Lagrange equation that holds the following:

$$\delta\mathcal{L} = \frac{\partial\mathcal{L}}{\partial\Psi}(\delta\Psi) + \sum_{j=0}^{3}\frac{\partial\mathcal{L}}{\partial(\partial_j\Psi)}\partial_j(\delta\Psi) = \sum_{j=0}^{3}\partial_j\left(\frac{\partial\mathcal{L}}{\partial(\partial_j\Psi)}\delta\Psi\right) = 0. \tag{A.27}$$

If the Lagrangian is invariant under infinitesimal transformations, then the equation of motion is invariant as well (the converse is not necessarily true). Suppose now that $\delta\Psi$ be the infinitesimal field variation of a continuous symmetry. We know that $\delta S = 0$ and let $\delta\Psi$ be a linear in the infinitesimal transformation parameters α; then by $\delta_\alpha\mathcal{L}$ we

denote the corresponding change of the Lagrangian. In order that the field equations be invariant, it is sufficient that the Lagrangian changes by a *total divergence*, i. e.,

$$\delta_\alpha \mathcal{L} = \sum_{j=0}^{3} \partial_j K_j(\Psi, \alpha) \tag{A.28}$$

for some set of functions $K_j(\Psi, \alpha)$, $j = 0, 1, 2, 3$, where Ψ generically denotes the fields on which Lagrangian depends. This follows from the fact that $\delta S = \int_{\Omega_4} \delta\mathcal{L} = \int_{\Omega_4} \nabla \cdot \overrightarrow{K} = \oint_{S_4} dS \overrightarrow{n}\overrightarrow{K} = 0$, i. e., vanishes on the boundary surface S_4 of the 4-dimensional time-space region Ω_4. However, the equation of motions describe the dynamics inside the integration domain, and the boundary conditions are separately imposed as independent conditions on the possible solutions (e. g., when in classical field theory we want to find the physical path between two fixed points by this variational principle, we fix the boundary conditions for these two endpoints and the computation of the extremum of the action is given by integration of the Lagrangian between these two endpoints). Based on this observation, under arbitrary infinitesimal variations $\delta\Psi$, the Lagrangian must change into the field variations $\delta\Psi$ multiplied by their corresponding field equations $E(\Psi)$, and a total derivative term of a function $N(\Psi, \delta\Psi)$ associated with the boundary contribution and linearly depending on $\delta\Psi$, so that

$$\delta\mathcal{L} = \delta\Psi E(\Psi) + \sum_{j=0}^{3} \partial_j N_j(\Psi, \delta\Psi). \tag{A.29}$$

Combining the equations (A.28) and (A.29), where we chose $\delta\Psi = \delta_\alpha\Psi$, we conclude that the divergence of $K_j(\Psi, \alpha) - N_j(\Psi, \delta_\alpha\Psi)$ must be proportional to the field equation $E(\Psi)$. Thus, by Noether's theorem, for any continuous symmetry, there exists the *Noether current* $J_j = N_j(\Psi, \delta_\alpha\Psi) - K_j(\Psi, \alpha)$ which, according to the above arguments satisfies $\sum_{j=0}^{3} \partial J_j = 0$. Thus, by taking the boundary component $N_j(\Psi, \delta_\alpha\Psi) = \frac{\partial\mathcal{L}}{\partial(\partial_j\Psi)}\delta\Psi$ from the last expression in the equation (A.27), we obtain in the general case when we have $n \geq 1$ Lagrangian independent variables $\Psi_i, 1 \leq i \leq n$,

$$J_j(\Psi, \alpha) = \sum_{i=1}^{n} \frac{\partial\mathcal{L}}{\partial(\partial_j\Psi_i)}\delta\Psi_i(\alpha) - K_j(\alpha), \tag{A.30}$$

and hence by elimination of the parameters in $\alpha = \{\alpha_i\}$ from this equation, for the particular definitions of $\delta\Psi_i(\alpha)$, we obtain a number of currents J_i^j by requiring that $\delta\mathcal{L} = 0$.

Finally, we obtain the continuity equation (a *locally conserved* current law)

$$\frac{\partial}{\partial t}J_0 + \sum_{i=1}^{3} \frac{\partial}{\partial q_i}J_i = \frac{\partial}{\partial t}J_0 + \nabla \cdot \overrightarrow{J} = 0 \tag{A.31}$$

where $\overrightarrow{J} = J_1\mathbf{e}_1 + J_2\mathbf{e}_2 + J_3\mathbf{e}_3$ is the Noether's current. Integrating equation (A.31) over all space $\mathbb{R}^3$, we obtain

$$0 = \int_{\mathbb{R}^3}\left(\frac{\partial}{\partial t}J_0 + \nabla\cdot\overrightarrow{J}\right)dV = \frac{d}{dt}\int_{\mathbb{R}^3} J_0 dV + \int_{\mathbb{R}^3}\nabla\cdot\overrightarrow{J}\,dV = \frac{d}{dt}\int_{\mathbb{R}^3} J_0 dV \tag{A.32}$$

because the second term can be converted into a surface integral (by Gauss' theorem) that vanishes in a sphere with enough big radius, from the fact that Ψ has a *finite* volume which is contained in such a big sphere.

Notice that in the field theory, the time-space quantity J_0 is called Noether's charge density, so that the Noether's charge (a *globally conserved* constant of motion) Q is defined by the integral over space of this charge density. That is, $Q = \int_{\mathbb{R}^3} J_0 dV$ so that from the equation (A.32) above, we have

$$\frac{dQ}{dt} = 0. \tag{A.33}$$

Analogously to the point mechanics, we have that the invariances of the Lagrangian density correspond to conservation laws. The general equation (A.30) for the Noether's current components is not easy to use for the computation of them. Thus, we can use the following method based on the one-parameter family (α) of transformations for which the Lagrangian $\mathcal{L}$ remains invariant $\Psi(t,\overrightarrow{\mathbf{r}}) \mapsto \widetilde{\Psi}(t,\overrightarrow{\mathbf{r}},\alpha)$, where α is the continuous parameter, such that $\widetilde{\Psi}(t,\overrightarrow{\mathbf{r}},0) = \Psi(t,\overrightarrow{\mathbf{r}})$ (i. e., for $\alpha = 0$ we obtain the identity transformation; such transformations are typical for the Lie groups of transformations). The transformations may be nonlinear in the generalized coordinates. Suppose that the Lagrangian $\mathcal{L}$ is invariant under replacement $\Psi \to \widetilde{\Psi}$ (i. e., invariant under this family (α) of one-parameter transformations). Then we must have

$$\begin{aligned} 0 &= \frac{d}{d\alpha}\Big|_{\alpha=0}\mathcal{L}(\widetilde{\Psi},\partial_j\widetilde{\Psi},q_j,v_j) = \left(\frac{\partial\mathcal{L}}{\partial\Psi}\frac{\partial\widetilde{\Psi}}{\partial\alpha} + \sum_{j=0}^{3}\frac{\partial\mathcal{L}}{\partial(\partial_j\Psi)}\frac{\partial_j\widetilde{\Psi}}{\partial\alpha}\right)_{\alpha=0} \\ &= \left(\sum_{j=0}^{3}\partial_j\frac{\partial\mathcal{L}}{\partial(\partial_j\Psi)}\frac{\partial\widetilde{\Psi}}{\partial\alpha} + \frac{\partial\mathcal{L}}{\partial(\partial_j\Psi)}\partial_j\frac{\partial\widetilde{\Psi}}{\partial\alpha}\right)_{\alpha=0} = \sum_{j=0}^{3}\partial_j\left(\frac{\partial\mathcal{L}}{\partial(\partial_j\Psi)}\frac{\partial\widetilde{\Psi}}{\partial\alpha}\right)_{\alpha=0}. \end{aligned}$$

This result can be obtained by substituting $\frac{\delta\Psi}{\delta\alpha}$ with $\frac{\partial\widetilde{\Psi}}{\partial\alpha}|_0$ and $\overrightarrow{K} = 0$ in equation (A.30). We can write this as $\sum_{j=0}^{3}\partial_j J_j = 0$, where

$$J_j = \frac{\partial\mathcal{L}}{\partial(\partial_j\Psi)}\frac{\partial\widetilde{\Psi}}{\partial\alpha}\Big|_{\alpha=0}, \quad \text{for } j = 0,1,2,3. \tag{A.34}$$

A.4.1 Vector fields on curved differentiable manifolds

It is necessary to consider the velocity vector field $\mathbf{W}$ of the particle's matter/energy density $\Phi_m(t, \overrightarrow{\mathbf{r}})$, where $(t, \overrightarrow{\mathbf{r}}) \in M^4$ is the Minkowski coordinate representation of a given point $p = \mathbf{r}_4$ of the curved 4-dimensional time-space manifold $\mathcal{M}$ generated by particle's energy-density Φ_m (in what follows, for simplicity we will use p instead of $\mathbf{r}_4$), so that the vector velocity in this point $p \in \mathcal{M}$ is given by $\mathbf{W}_p \equiv \mathbf{W}(p)$. This more general (than Minkowski's) framework for vector fields will be introduced, based mainly on the books on differentiable manifolds [169–171].

Let us denote by $C^k(\mathcal{M})$ the subset of functions $f : \mathcal{M} \to \mathbb{R}$, which are k-differentiable, for $k \gg 1$ (here we consider *smooth* case when $k = \infty$). The important subset in $C^k(\mathcal{M})$ are the four coordinate functions or, simply *coordinates*, $q_i : \mathcal{M} \to \mathbb{R}$, $0 \leq i \leq 3$. Roughly speaking, a n-dimensional manifold $\mathcal{M}$ is a set of points that can be labeled by coordinates, $\phi : \mathcal{M} \to \mathbb{R}^n$, and here we will consider the smooth curved 4-dimensional time-space manifolds. For a given (local) chart (also called a *local coordinate system* on $\mathcal{M}$), (U, ϕ), where $U \subset \mathcal{M}$ with one-to-one invertible (continuous in both directions) map $\phi : U \to \mathbb{R}^4$ to some open subset of the Euclidean 4-D space $\mathbb{R}^4$ (with its inverse ϕ^{-1}), for any point $p \in U$, $q_i(p)$ is the value of its i-th coordinate, such that $\phi(p) = (q_0(p), \ldots, q_3(p))$, where $q_0(p)$ is the time coordinate, while other three are the space coordinates of this point p of the curved 4-dimensional manifold $\mathcal{M}$ (curved time space in GR). The fact that ϕ is one-to-one mapping ensures that two different points of U differ, at least, in the value of one of the coordinates.

Analogously, for any other (local) coordinate system of chart (U', ϕ'), different from this (U, ϕ) chart coordinate system, we will denote by $q'_j \in C^k(\mathcal{M})$ this new coordinate system. These two charts are C^k-compatible if $U \bigcap U' = \emptyset$ (the empty set), or if the *homeomorphisms*[4] (also known as a change or transformation of coordinates or *transition maps*) $\phi \circ \phi'^{-1} : \phi'(U \bigcap U') \to \phi(U \bigcap U')$ and $\phi' \circ \phi^{-1} : \phi(U \bigcap U') \to \phi'(U \bigcap U')$, whose domains are open in $\mathbb{R}^4$, are differentiable of class C^k. That is, for all points $p \in U \bigcap U'$, $q_0(p), \ldots, q_3(p)$ be differentiable functions of class C^k (have all k-th continuous partial derivatives) of $q'_0(p), \ldots, q'_3(p)$, and conversely.

Differentiable manifold

We assume that this property is satisfied and that the coordinate changes (the two homomorphisms above) are also *diffeomorphisms*,[5] for any two charts (thus for any atlas) of the continuous differentiable and smooth curved time-space manifold $\mathcal{M}$ used in general relativity. This manifold inherits many of the local properties of Euclidean 4-D space: it is locally path connected, locally compact and locally metrizable. In other words, this

4 A bijective continuous function whose inverse is also continuous.

5 Invertible maps that are smooth (of class C^∞) in both directions.

differentiable (or, smooth) manifold $\mathcal{M}$ is a topological manifold with a globally defined differentiable (or smooth) structure. Each point p of differentiable manifold $\mathcal{M}$ has a tangent space, which is a flat Minkowski (pseudo-Euclidean) time-space $\mathbb{R}^{1,3}$ consisting of the tangent vectors of the curves through the point. We recall that a Riemannian manifold is an analytic manifold in which each tangent space is equipped with an inner (or scalar) product of tangent vectors, in a manner which varies smoothly from point to point. This allows one to define various notions such as length, angles, areas (or volumes), curvature, gradients of functions and divergence of vector fields. Any smooth manifold admits a Riemannian metric, and hence also the pseudo-Riemannian 4-D differentiable smooth manifold $\mathcal{M}$ used in the general relativity theory. In fact, in this pseudo-Riemannian manifold $\mathcal{M}$, the covariant derivative is often used for the Levi–Civita connection given by Christoffel symbols Γ^j_{ik}.

Any differentiable curve lying in the subset U of the manifold $\mathcal{M}$ can be given by a mapping $\gamma : I \to U \subset \mathcal{M}$, (we denote by $I \subset \mathbb{R}$ the open subset of $\mathbb{R}$), so that for each $s \in \mathbb{R}$, we obtain some point $p = \gamma(s) \in \mathcal{M}$.

The *tangent vector* to γ at the point $p = \gamma(s_0)$, denoted by the mapping $\gamma'_{s_0} : C^k(\mathcal{M}) \to \mathbb{R}$, is defined for any function $f : \mathcal{M} \to \mathbb{R}$ by

$$\gamma'_{s_0}(f) \triangleq \frac{d}{ds}(f \circ \gamma)\Big|_{s_0} = \lim_{s \to s_0} \frac{f(\gamma(s)) - f(\gamma(s_0))}{s - s_0} \tag{A.35}$$

where $f \circ \gamma$ denotes the composition of these two mappings. Thus, this vector is tangent vector on the curve γ at point $p = \gamma(s_0)$, and is obtained as a derivation of function f along this curve γ. The 4-dimensional tangent space to the manifold $\mathcal{M}$ at point p, denoted by T_pM, is the set of all vectors tangent to $\mathcal{M}$ at p. For a chart (U, ϕ) on $\mathcal{M}$, with coordinates (functions) $q_i : \mathcal{M} \to \mathbb{R}$ and point $p \in U$, the *natural* (or *canonical*) vector basis of this local coordinate system at point p is defined by vectors $(\frac{\partial}{\partial q_i})_p \in T_pM$ for $0 \le i \le 3$, so that for each mapping $f \in C^k(\mathcal{M})$ such that, for the function $f \circ \phi^{-1} : \mathbb{R}^4 \to \mathbb{R}$, $\frac{\partial}{\partial q_i}(f \circ \phi^{-1})$, is the partial derivative w. r. t. the i-th argument of this function,

$$\begin{aligned}\left(\frac{\partial}{\partial q_i}\right)_p [f] &\triangleq \frac{\partial}{\partial q_i}(f \circ \phi^{-1})\Big|_{\phi(p)} \\ &= \lim_{s \to 0} \frac{f(\phi^{-1}(q_0(p), \dots, q_i(p) + s, \dots, q_3(p))) - f(\phi^{-1}(q_0(p), \dots, q_i(p), \dots, q_3(p)))}{s}\end{aligned} \tag{A.36}$$

and for s sufficiently small, so that all the points belong to U, we obtain that for $f = q_j$, $(\frac{\partial}{\partial q_i})_p(q_j) = \frac{\partial q_j}{\partial q_i}|_p = \delta_{ij}$, and hence we obtained the *natural vector basis*

$$\mathbf{e}_i \triangleq \left(\frac{\partial}{\partial q_i}\right)_p, \quad i = 0, 1, 2, 3 \tag{A.37}$$

of a locally flat Minkowski time-space corresponding to T_pM in a small region around the point p. Hence, this natural vector basis defines the components $g_{ij} \equiv \mathbf{e}_i \cdot \mathbf{e}_j = (\frac{\partial}{\partial q_i})_p \cdot (\frac{\partial}{\partial q_j})_p$ of the *metrics* (tensor field identified as the gravitational potential) at a point p:

$$\mathbf{g}_{jk} = \begin{pmatrix} g_{00} & g_{01} & g_{02} & g_{03} \\ g_{10} & g_{11} & g_{12} & g_{13} \\ g_{20} & g_{21} & g_{22} & g_{23} \\ g_{30} & g_{31} & g_{32} & g_{33} \end{pmatrix}. \tag{A.38}$$

The connection coefficients Γ^j_{ik}, also known as Christoffel symbols of the *second kind*, are introduced by

$$\frac{\partial}{\partial q_k}\mathbf{e}_i \triangleq \sum_j \Gamma^j_{ik}\mathbf{e}_j \tag{A.39}$$

for the vector basis $\mathbf{e}_i$, where $\Gamma^j_{ik} = \Gamma^j_{ki}$ in nontorsion case that is only considered here. The Christoffel symbols in terms of the metric can be obtained by introduction of Christoffel symbols of the *first kind*,

$$\begin{aligned}\Gamma_{lik} = \Gamma_{lki} \equiv \sum_m g_{lm}\Gamma^m_{ik}, \quad \text{from } \frac{\partial}{\partial q_k}g_{il} = \frac{\partial}{\partial q_k}(\mathbf{e}_i \cdot \mathbf{e}_l) = \mathbf{e}_i\frac{\partial}{\partial q_k}\mathbf{e}_l + \mathbf{e}_l\frac{\partial}{\partial q_k}\mathbf{e}_i = \text{from (A.39)} \\ = \sum_j (g_{ij}\Gamma^j_{lk} + g_{lj}\Gamma^j_{ik}) = \Gamma_{ilk} + \Gamma_{lik},\end{aligned}$$

and hence,

$$\begin{aligned}\Gamma_{lik} = \frac{1}{2}\left(\frac{\partial}{\partial q_k}g_{li} + \frac{\partial}{\partial q_i}g_{lk} - \frac{\partial}{\partial q_l}g_{ik}\right), \quad \text{that is,} \\ \sum_l g^{jl}\Gamma_{lik} = \sum_l g^{jl}\left(\sum_m g_{lm}\Gamma^m_{ik}\right) = \sum_m\left(\sum_l g^{jl}g_{lm}\right)\Gamma^m_{ik} = \sum_m \delta_{jm}\Gamma^m_{ik} = \Gamma^j_{ik},\end{aligned}$$

i. e.,

$$\Gamma^j_{ik} = \frac{1}{2}\sum_l g^{jl}\left(\frac{\partial}{\partial q_k}g_{li} + \frac{\partial}{\partial q_i}g_{lk} - \frac{\partial}{\partial q_l}g_{ik}\right) \tag{A.40}$$

and, consequently, $\Gamma^j_{ik} = \Gamma^j_{ki}$, and

$$\sum_j \Gamma^j_{jk} = \sum_j \Gamma^j_{kj} = \frac{1}{2}\sum_{i,j} g^{ij}\frac{\partial}{\partial q_k}g_{ij}. \tag{A.41}$$

Based on them are introduced the Riemann curvature tensor R^j_{ink} (it has 20 degrees of freedom), in the nontorsion case, by

$$R^{j}_{ink} \triangleq \frac{\partial}{\partial q_n}\Gamma^{j}_{ik} - \frac{\partial}{\partial q_k}\Gamma^{j}_{\text{in}} + \sum_{m}(\Gamma^{j}_{mn}\Gamma^{m}_{ik} - \Gamma^{j}_{mk}\Gamma^{m}_{\text{in}}). \tag{A.42}$$

The reduction from the curvilinear Riemann space of GR into a locally flat Minkowski space is given by the condition $R^{j}_{ink} = 0$. By tensor contraction, we define the tensors

$$R_{ik} \triangleq \sum_{j} R^{j}_{ijk}, \tag{A.43}$$

and hence the well-known Ricci scalar

$$R \triangleq \sum_{ik} g^{ik} R_{ik}. \tag{A.44}$$

The covariant derivatives of metric tensors are an example of the conservation of the time-space metric invariance. For the flat Minkowski time space with the metric tensor (η_{ij}) given by equation (A.22), the flat metric property is invariant at each time point, i. e., $\frac{\partial}{\partial q_k}\eta_{ij} = 0$ for all components of this tensor. But in curved time space, we have that generally this invariance is not preserved by ordinary derivatives, i. e., from (A.39), $\frac{\partial}{\partial q_k} g_{ij} = \frac{\partial}{\partial q_k}(\mathbf{e}_i \mathbf{e}_j) = \mathbf{e}_i \frac{\partial}{\partial q_k}\mathbf{e}_j + \mathbf{e}_j \frac{\partial}{\partial q_k}\mathbf{e}_i = \mathbf{e}_i \sum_n \Gamma^{n}_{jk}\mathbf{e}_n + \mathbf{e}_j \sum_n \Gamma^{n}_{ik}\mathbf{e}_n = \sum_n \Gamma^{n}_{jk}(\mathbf{e}_i\mathbf{e}_n) + \sum_n \Gamma^{n}_{ik}(\mathbf{e}_n\mathbf{e}_j) = \sum_n \Gamma^{n}_{jk} g_{\text{in}} + \sum_n \Gamma^{n}_{ik} g_{nj} \neq 0$. But we can see that this invariance is preserved by the covariant derivatives D_k of curved time-space metrics, i. e., $D_k g_{ij} = 0$, so that the curved time-space metric is invariant under covariant derivations.

Thus, any tangent 4-vector $\mathbf{v}_p : C^k(\mathcal{M}) \to \mathbb{R}$ on the manifold at point p, i. e., $\mathbf{v}_p \in T_pM$ (for simplicity, for the vectors $\mathbf{e}_j$ in a vector basis we will avoid to use the "arrow" subscript $\rightarrow$), can be expressed as a linear composition of these base vectors by

$$\mathbf{v}_p = \sum_{i=0}^{3} \mathbf{v}_p(q_i)\left(\frac{\partial}{\partial q_i}\right)_p. \tag{A.45}$$

In effect, we have that for each coordinate $q_j \in C^k(\mathcal{M})$, is satisfied

$$\mathbf{v}_p(q_j) = \sum_{i=0}^{3} \mathbf{v}_p(q_i)\left(\frac{\partial}{\partial q_i}\right)_p (q_j) = \sum_{i=0}^{3} \mathbf{v}_p(q_i)\delta_{ij}.$$

Thus, the tangent vector to curve γ at the point $p = \gamma(s_0)$ can be expressed as a linear combination of the basic vectors as $\gamma'_{s_0} = \sum_{i=0}^{3} \gamma'_{s_0}(q_i)(\frac{\partial}{\partial q_i})_{\gamma(s_0)} = \sum_{i=0}^{3} \frac{d(q_i \circ \gamma)}{ds}|_{s_0}(\frac{\partial}{\partial q_i})_{\gamma(s_0)}$.

A *geodesic* in $\mathcal{M}$ is a curve $\gamma : I \to \mathcal{M}$ whose acceleration $\ddot{\gamma}$ is everywhere orthogonal to $\mathcal{M}$. Thus, a geodesic is a curve in $\mathcal{M}$, which always goes "straight ahead" in the surface. Its acceleration serves only to keep it in the surface and it has no component of acceleration tangent to the surface, so that it has a constant speed $\dot{\gamma}(s) = \frac{d\gamma}{ds}$ (in this case of geodesic, the parameter s represents the time).

A vector field $\mathbf{W}$ on $U \subseteq \mathcal{M}$ is a function $\mathbf{W} : U \to TM$ that to each point p of U assigns a tangent vector $\mathbf{W}_p \equiv \mathbf{W}(p) \in T_p\mathcal{M}$. For a given chart (U, ϕ), we have four differentiable vector fields $(\frac{\partial}{\partial q_i})$ on U defined by $(\frac{\partial}{\partial q_i})(p) \equiv (\frac{\partial}{\partial q_i})_p$.

Since the tangent vectors $(\frac{\partial}{\partial q_i})_p$ form a vector basis on T_pM, any vector field $\mathbf{W}$ evaluated at point $p \in U$ must be linear combination of the tangent vectors $(\frac{\partial}{\partial q_i})_p$ with real coefficients $W_i(p) \in \mathbb{R}$, so that $\mathbf{W}(p) = \sum_i W_i(p)(\frac{\partial}{\partial q_i})_p = \sum_i W_i(p)(\frac{\partial}{\partial q_i})(p) = \sum_i [W_i(\frac{\partial}{\partial q_i})](p)$, with functions $W_i : U \to \mathbb{R}$ such that for each $p \in U$, $W_i(p) \equiv \mathbf{W}_p(q_i)$, so that this vector field $\mathbf{W}$ is locally given by

$$\mathbf{W} = \sum_{i=0}^{3} W_i\Big(\frac{\partial}{\partial q_i}\Big), \tag{A.46}$$

and hence the vector field $\mathbf{W}$ is differentiable iff the functions W_i are. Let $(q_0, \ldots, q_3)$ and $(q'_0, \ldots, q'_3)$ be two coordinate systems in U and U', respectively. In tensor formalism, $\mathbf{W}$ is a *contravariant* vector.

From (A.46), any vector field in the 4-dimensional manifold M can be given in terms of any set of 4-vector fields such that at each point $p \in M$ of their common domain forms a basis $\{(\frac{\partial}{\partial q_i})_p \mid 0 \leq i \leq 3\}$ for the tangent space T_pM at this point p. The use of this kind of set of vector fields, not necessarily associated with coordinate systems, may be convenient when there is some additional structure on the manifold (e. g., a connection, a *metric tensor* $\mathbf{g}_{jk}$, or a Lie group structure like Poincare transformations).

A.4.2 Transformation of coordinates

Suppose that we have two sets of curvilinear coordinates $(q_0, \ldots, q_3)$ and $(q'_0, \ldots, q'_3)$ then a general 4×4 transformation (i. e., a nonlinear map) $q_i \mapsto q'_i$ is defined by the set of transformation equations

$$q'_i = q'_i(q_0, \ldots, q_3), \quad i = 0, 1, 2, 3 \tag{A.47}$$

which defines the Jacobian matrix $\mathbf{J}' \equiv (\frac{\partial q'_i}{\partial q_j})$, with an example given by (A.50), so that the transformation of vector basis is given by $[\mathbf{e}_j] = \mathbf{J}'^{\,T}[\mathbf{e}'_j]$, where $[\mathbf{e}'_j]$ and $[\mathbf{e}_j]$ are the columns of the basis vectors for these two coordinate systems.

Now, if the Jacobian determinant of this coordinate transformation is different from zero

$$J' = \Big|\frac{\partial q'_i}{\partial q_j}\Big| \neq 0 \tag{A.48}$$

then the transformation (A.47) is reversible and the inverse coordinate transformation

$$q_j = q_j(q'_0, \ldots, q'_3), \quad i = 0, 1, 2, 3 \tag{A.49}$$

exists as well with the Jacobian determinant of this inverse coordinate transformation $J = |\frac{\partial q_i}{\partial q'_j}| \neq 0$. Finding the inverse transformation is the problem of matrix inverse. For example, in the Minkowski time space, we can transform the Cartesian coordinates $(q'_0, \ldots, q'_3) = (ct, x, y, z)$ into spherical coordinates $(q_0, \ldots, q_3) = (ct, r, \theta, \varphi)$, with coordinate transformations, $q'_0 = q_0$, $q'_1 = r\cos\theta\cos\varphi = q_1\cos q_2\cos q_3$, $q'_2 = r\sin\theta\cos\varphi = q_1\sin q_2\cos q_3$, and $q'_3 = r\sin\varphi = q_1\sin q_3$, with the Jacobian matrix given by

$$\mathbf{J}' \equiv \left(\frac{\partial q'_i}{\partial q_j}\right) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos q_2\cos q_3 & -q_1\sin q_2\cos q_3 & q_1\sin q_2\sin q_3 \\ 0 & \sin q_2\cos q_3 & q_1\cos q_2\cos q_3 & -q_1\sin q_2\sin q_3 \\ 0 & \sin q_3 & 0 & q_1\cos q_3 \end{pmatrix} \tag{A.50}$$

and the corresponding Jacobian determinant $J' = |\mathbf{J}'| = |\frac{\partial q'_i}{\partial q_j}| = (q_1)^2\cos q_3$. An inverse coordinate transform is given by $q_0 = q'_0$, $q_1 = \sqrt{(q'_1)^2 + (q'_2)^2 + (q'_3)^2}$, $q_2 = \arctan(\frac{q'_2}{q'_1})$, and $q_3 = \arctan(\frac{q'_3}{\sqrt{(q'_1)^2+(q'_2)^2}})$, with $J = |\frac{\partial q_i}{\partial q'_j}| = \frac{1}{(q_1)^2\cos q_3}$.

The *covariant* metric tensor, for coordinate transformation (A.47), transforms as

$$g_{jk} = \mathbf{e}_j \cdot \mathbf{e}_k = \left(\sum_l \mathbf{e}'_l \frac{\partial q'_l}{\partial q_j}\right)\left(\sum_m \mathbf{e}'_m \frac{\partial q'_m}{\partial q_k}\right) = \sum_{l,m=0}^{3} \frac{\partial q'_l}{\partial q_j}\frac{\partial q'_m}{\partial q_k} g'_{lm}, \tag{A.51}$$

and hence g_{jk} is the element of the matrix obtained by multiplication of the j-th row of the transposed Jacobian matrix $\mathbf{J}'^{\,T} = (\frac{\partial q'_i}{\partial q_j})^T$ and k-th column of the matrix product $\mathbf{g}'_{jk} \cdot (\frac{\partial q'_i}{\partial q_j}) = \mathbf{g}'_{jk} \cdot \mathbf{J}'$, i. e., $\mathbf{g}_{jk} = \mathbf{J}'^{\,T} \cdot (\mathbf{g}'_{jk} \cdot \mathbf{J}')$, and hence taking determinant of both sides of this matrix equation gives $g = J'^{\,2} g'$ (where $g = \det(\mathbf{g}_{jk}) < 0$), or

$$\sqrt{-g} = J'\sqrt{-g'} \tag{A.52}$$

where $J' = |\frac{\partial q'_j}{\partial q_k}|$ is the Jacobian determinant of the Jacobian matrix $\mathbf{J}'$ of the transformation. The infinitesimal 4-volume $d\Omega \equiv d^4q$ transforms as

$$d\Omega = \left|\frac{\partial q_j}{\partial q'_k}\right| d\Omega' = \frac{1}{J'} d\Omega'. \tag{A.53}$$

Consequently, the combination $\sqrt{-g}d\Omega$ transforms

$$\sqrt{-g}d\Omega = J'\sqrt{-g'}\frac{1}{J'}d\Omega' = \sqrt{-g'}d\Omega' \tag{A.54}$$

and is thus generally invariant under coordinate transformations, and hence the covariant infinitesimal time-space volume, used in (covariant) tensor equations, is of the form $\sqrt{-g}d\Omega \equiv d^4q\sqrt{-g}$.

Furthermore, from the fact that $\mathrm{Tr}(M^{-1}\frac{\partial M}{\partial q_k}) = \frac{\partial}{\partial q_k}\ln(\det M)$ (holds for matrices $n\times n$ of arbitrary size; in our case of 4×4 matrices), we have for $M = \mathbf{g}_{jk}$,

$$\begin{aligned}\frac{1}{2}\frac{\partial}{\partial q_k}\ln(\det(\mathbf{g}_{jk})) &= \frac{1}{2}\frac{\partial}{\partial q_k}\ln(-g) = \frac{\partial}{\partial q_k}\ln(\sqrt{-g}) = \frac{1}{\sqrt{-g}}\frac{\partial}{\partial q_k}\sqrt{-g} = \frac{1}{2}\mathrm{Tr}\left(\mathbf{g}^{-1}_{jk}\frac{\partial}{\partial q_k}\mathbf{g}_{jk}\right)\\ &= \frac{1}{2}\sum_j\left(\sum_i g^{ji}\frac{\partial g_{ij}}{\partial q_k}\right) = \frac{1}{2}\sum_{j,i} g^{ji}\frac{\partial g_{ij}}{\partial q_k},\end{aligned}$$

and hence from (A.41), and the fact that metric tensor $\mathbf{g}_{jk}$ is symmetric, and hence also its inverse tensor is symmetric, we obtain

$$\sum_j \Gamma^j_{jk} = \frac{1}{2}\sum_{i,j} g^{ij}\frac{\partial g_{ij}}{\partial q_k} = \frac{1}{\sqrt{-g}}\frac{\partial}{\partial q_k}\sqrt{-g}, \quad \text{and} \quad \delta\sqrt{-g} = \frac{\sqrt{-g}}{2}\sum_{i,j} g^{ij}\delta g_{ij}. \tag{A.55}$$

Bibliography

[1] B. Russel, “Mysticism and Logic: And other essays,” *Pearson Longman*, 1919.

[2] A. Korzybski, “Science and Sanity: An Introduction of Non-Aristotelian Systems and General Semantics,” *Institute of General Semantics, 5th edition, ISBN 0-937298-01-8*, 1994.

[3] J. B. Shaw, “Lectures on the Philosophy of Mathematics,” *Chicago, London: The Open Court Publishing Company*, 1918.

[4] Z. Majkić, “Categories: symmetry, n-dimensional levels and applications,” *PhD Thesis, University “La Sapienza”, Roma, Italy*, 1998.

[5] Z. Majkić, “Big Data Integration Theory,” *Texts in Computer Science, New York Inc.: Springer-Verlag, ISBN: 978-3-319-04156-8*, 2014.

[6] S. Dehaene and E. M. Brannon, “Space, Time and Number in the Brain: Searching for the Foundations of Mathematical Thought,” *London: Elsevier*, 2011.

[7] Z. Majkić, “Intensional First Order Logic: from AI to New SQL Big Data,” *Berlin/Boston: Walter De Gruyter GmbH, ISBN 978-3-11-099494-0*, 2022.

[8] Z. Majkić, “Completion and Unification of Quantum Mechanics with Einstein’s GR Ideas, Part I: Completion of QM,” *New York: Nova Science Publishers, ISBN 978-1-53611-946-6, July*, 2017.

[9] Z. Majkić, “Completion and Unification of Quantum Mechanics with Einstein’s GR Ideas, Part II: Unification with GR,” *New York: Nova Science Publishers, ISBN 978-1-53611-947-3, September*, 2017.

[10] Z. Majkić, “Completion and Unification of Quantum Mechanics with Einstein’s GR Ideas, Part III: Advances, Revisions and Conclusions,” *New York: Nova Science Publishers, ISBN 978-1-53617-200-3, January*, 2020.

[11] A. Asperti and G. Longo, “Categories, Types and Structures,” *MIT Press*, 1991.

[12] S. Mac Lane, “Categories for the Working Mathematician,” *Springer-Verlag*, 1971.

[13] O. Udrea, V. S. Subrahmanian, and Z. Majkić, “Probabilistic RDF,” In *IEEE Conference on Information Reuse and Integration (IEEE IRI 2006), September 16–18, Waikoloa, Hawaii, USA*, 2006.

[14] Z. Majkić, O. Udrea, and V. S. Subrahmanian, “Aggregates in generalized temporally indeterminate databases,” In *Int. Conference on Scalable Uncertainty Management (SUM 2007), October 10-12, Washington DC, USA, LNCS 4772*, pp. 171–186, 2007.

[15] Z. Majkić, “Temporal probabilistic logic programs: state and revision,” In *International Conference in Artificial Intelligence and Pattern Recognition (AIPR-07), July 9–12, 2007, Orlando, FL, USA*, 2007.

[16] F. W. Lawvere, “Functorial semantics of algebraic theories,” *Proc. Natl. Acad. Sci., 50*, pp. 869–872, 1963.

[17] A. Noether, “Invariante variationsprobleme,” In *Göttinger Nachrichten, presented by F. Klein at the meeting of 26 July 1918*, pp. 235–257, 1918.

[18] D. Prawitz, “Ideas and results in proof theory,” In *Proceedings of the Second Scandinavian Logic Symposium*, 1971.

[19] R. Goldblat, “Topoi: The Categorial Analysis of Logic,” *Amsterdam: North-Holland*, 1979.

[20] A. Church, “An unsolvable problem of elementary number theory,” *Am. J. Math., 58*, pp. 354–363, 1936.

[21] H. B. Curry, “Functionality in combinatory logic,” *Proc. Natl. Acad. Sci. USA, 20*, pp. 584–590, 1934.

[22] A. Church, “A formulation of the simple theory of types,” *J. Symb. Log., 5*, pp. 56–68, 1940.

[23] N. G. de Bruijn, “The mathematical language AUTOMATH, its usage and some of its extensions,” In *M. Laudet, D. Lacombe and M. Schuetzenberger (Eds.), Symposium on Automatic Demonstration, INRIA, Versailles, Lecture Notes in Computer Science 125, Berlin: Springer-Verlag*, 1970.

[24] W. A. Howard, “The formulae-as-types notion of construction,” In *Hindley and Seldin*, pp. 479–490, 1980.

[25] H. B. Curry and R. Feys, “Combinatory Logic, vol. i,” *Amsterdam: North-Holland*, 1958.

[26] J. Lambek and P. Scott, “Introduction to Higher Order Categorial Logic,” *Cambridge University Press*, 1986.

https://doi.org/10.1515/9783111081670-009

[27] J. Lambek, “From lambda calculus to Cartesian closed categories,” In *J. P. Seldin and J. R Hindley (Eds.), To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism, New York: Academic Press*, pp. 375–402. 1980.

[28] D. S. Scott, “Relating theories of lambda calculus,” In *J. P. Seldin and J. R Hindley (Eds.), To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism, New York: Academic Press*, pp. 403–450. 1980.

[29] C. P. J. Koymans, “Models of lambda calculus,” *Inf. Control, 52(3)*, pp. 306–332, 1982.

[30] P. DiGianantonio, G. Franco, and F. Honsel, “Game semantics for untyped lambda calculus,” In *Proc. of the Conference Typed Lambda Calculus and Applications, LNCS 1591*, pp. 114–128. 1999.

[31] A. D. Ker, H. Nickau, and C. H. Ong, “A universal innocent game model for the b.”

[32] M. Karoubi, “K-theory, An introduction,” *Berlin/New York: Springer*, 1978.

[33] M. F. Atiyah, “K-theory,” *New-York: Benjamin*, 1967.

[34] A. Borel and J. P. Serre, “Le théorème de riemann-roch (d’après grothendieck),” *Bull. Soc. Math. Fr., 86*, pp. 97–136, 1958.

[35] T. Búhler, “Exact categories,” *Expo. Math., 28(1)*, pp. 1–69, 2010.

[36] F. Borceux and B. Dejean, “Cauchy completion in category theory,” In *Cahiers de topologie et géométrie différentielle catégoriques, tome 27, n. 2*, pp. 133–146, 1986.

[37] R. Milner, “A Calculus of Communicating Systems,” *LNCS 92, Springer*, 1980.

[38] R. Milner, “Communication and Concurrency,” *Prentice Hall*, 1989.

[39] M. Hennessy and R. Milner, “Algebraic laws for nondeterminism and concurrency,” *J. ACM, 67*, pp. 137–161, 1985.

[40] R. Milner, J. Parrow, and D. Walker, “A calculus of mobile processes,” *Inf. Comput., 100*, 1992.

[41] D. Park, “Concurrency and automata on infinite sequences,” In *Lecture Notes in Computer Science, 154, Springer*, pp. 561–572. 1981.

[42] J. A. Bergstra and J. W. Klop, “Algebra of communicating processes with abstraction,” *Theor. Comput. Sci., 37(1)*, pp. 77–121, 1985.

[43] G. Winskel, “Synchronization trees,” *Theor. Comput. Sci., 34*, pp. 33–82, 1984.

[44] S. Kasangian and A. Labella, “Enriched categorical semantics for distributed calculi,” *J. Pure Appl. Algebra, 83(3)*, pp. 295–321, 1992.

[45] J. Bénabou, “Introduction to Bicategories,” *LNM 47, Berlin: Springer*, pp. 1–77, 1967.

[46] G. M. Kelly and R. H. Street, “Review of the Elements of 2-Categories,” *LNM 420, Berlin: Springer*, pp. 75–103. 1975.

[47] G. Winskel, “Category theory and models of parallel computations,” In *Category Theory and Computer Programming*, 1985.

[48] S. Kasangian, A. Labella, and A. Pettorossi, “Observers, experiments and agents: a comprehensive approach to parallelism,” In *Semantics of Concurrency, LNCS 469, Springer*, pp. 375–406, 1990.

[49] P. Aczel and N. Mendler, “A final coalgebra theorem,” In *Proc. Category Theory and Computer Science, LNCS 389, Springer*, pp. 357–365, 1989.

[50] J. Van Benthem, “Correspondence Theory,” In *D. Gabbay and F. Guenthner (Eds.), Handbook of Philosophical Logic, Vol.II, Holland: D. Reidel Publishing Company*, pp. 167–247, 1984.

[51] J. Van Benthem, “A new modal Lindström theorem,” *Log. Univers., 1*, pp. 125–148, 2007.

[52] J. Groote, “Transition system specifications with negative premises,” *Theor. Comput. Sci., 118*, pp. 263–299, 1993.

[53] D. Kozen, “Results on the propositional mu-calculus,” *Theor. Comput. Sci., 27*, pp. 333–354, 1983.

[54] P. Aczel, “Non-Well-Founded Sets,” Lecture Notes CSLIm 14, 1988.

[55] P. Aczel, “Final universes of processes,” In *S. Brookes et al. (Eds.), Proc. Mathematical Foundations of Programming Semantics (MFPS’93), Lecture Notes in Computer Science 802, Springer*, 1993.

[56] J. Rutten and D. Turi, “Initial Algebra and Final Coalgebra Semantics for Concurrency,” *LNCS 803, Springer*, pp. 530–582, 1994.

[57] B. Jacobs and J. Rutten, "A tutorial on (co)algebras and (co)induction," *Bull. Eur. Assoc. Theor. Comput. Sci., 62*, pp. 222–259, 1997.

[58] Z. Majkić, "The category-theoretic semantics for database mappings," *Technical Report 14-03, University 'La Sapienza', Roma, Italy*, 2003.

[59] Z. Majkić, "Fixpoint semantics for query answering in data integration systems," In *AGP03–8.th Joint Conference on Declarative Programming, Reggio Calabria*, pp. 135–146, 2003.

[60] Z. Majkić, "Coalgebraic semantics for logic programming," In *18th Workshop on (Constraint) Logic Programming, W(C)LP 2004, March 04-06, Berlin, Germany*, 2004.

[61] Z. Majkić, "Abstract database category based on relational-query observations," In *International Conference on Theoretical and Mathematical Foundations of Computer Science (TMFCS-08), Orlando FL, USA, July 7-9*, 2008.

[62] Z. Majkić, "Truth and knowledge fixpoint semantics for many-valued logic programming," In *19th Workshop on (Constraint) Logic Programming (W(C)LP 2005), February 21-25, Ulm, Germany*, 2005.

[63] Z. Majkić, "Observational logic for composition of agents," In *International Conference on Artificial Intelligence and Pattern Recognition (AIPR-07), 9-12 July, Orlando FL, USA*, 2007.

[64] Z. Majkić, "Induction principle in relational database category," In *Int. Conference on Theoretical and Mathematical Foundations of Computer Science (TMFCS-09), Orlando FL, USA, July 13-16*, 2009.

[65] Z. Majkić, "Coalgebraic specification of query computation in intensional P2P database systems," In *Int. Conference on Theoretical and Mathematical Foundations of Computer Science (TMFCS-08), Orlando FL, USA, July 7-9*, 2008.

[66] Z. Majkić and B. Prasad, "Kleisli category for database mappings," *Int. J. Intell. Inf. Database Syst., 4(5)*, pp. 509–527, 2010.

[67] Z. Majkić, "Data base mappings and monads: (co)induction," *arXiv:1102.4769v1, 22 February*, pp. 1–31, 2011.

[68] Z. Majkić, "Autoreferential semantics for many-valued modal logics," *J. Appl. Non-Class. Log., 18(1)*, pp. 79–125, 2008.

[69] R. Suszko, "Remarks on Lukasiewicz's three-valued logic," *Bull. Sect. Log., 4*, pp. 87–90, 1975.

[70] D. Batens, "A bridge between two-valued and many-valued semantic systems: n-tuple semantics," In *Proc. of the XII ISMVL*, pp. 318–322, 1982.

[71] D. Scott, "Completeness and axiomatizability in many-valued logic," In *Proc. of Tarski Symposium, Proc. of Symposia in Pure Mathematics 25, Berkeley*, pp. 411–436, 1974.

[72] N. C. A. da Costa, J. Y. Béziau, and O. A. S. Bueno, "Malinowski and Suszko on many-valuedness: on the reduction of many-valuedness to two-valuedness," *Mod. Log., 6*, pp. 272–299, 1996.

[73] C. Caleiro, W. A. Carnielli, M. E. Coniglio, and J. Marcos, "Two's company: the humbug of many logical values," In *J. Y. Béziau (Ed.), Logica Universalis, Birkhauser Verlag*, pp. 169–189, 2005.

[74] M. C. Fitting, "Bilattices are nice things," In *Proceedings of Conference on Self-Reference, Copenhagen, Denmark*, 2002.

[75] N. D. Belnap, "A useful four-valued logic," In *J-M. Dunn and G. Epstein (Eds.), Modern Uses of Multiple-Valued Logic, D. Reidel*, 1977.

[76] K. Došen, "Negation as a modal operator," *Rep. Math. Log., 20*, pp. 15–28, 1986.

[77] J. Y. Béziau, "Are paraconsistent negations negations?" In *W. Carnielli (Ed.), Paraconsistency: The Logical Way to the Inconsistent, New York: Marcel Dekker*, pp. 465–486, 2002.

[78] Z. Majkić, "Paraconsistent da Costa weakening of intuitionistic negation: what does it mean?" *Int. J. Pure Math. – NAUN, 9*, pp. 35–48, 2022.

[79] M. Ginsberg, "Multivalued logics: a uniform approach to reasoning in artificial intelligence," *Comput. Intell., 4*, pp. 265–316, 1988.

[80] Z. Majkić, "Autoepistemic logic programming for reasoning with inconsistency," In *International Symposium on Logic-based Program Synthesis and Transformation (LOPSTR), September 7-9, 2005, Imperial College, London, UK*, 2005.

[81] M. C. Fitting, “Bilattices and the semantics of logic programming,” *J. Log. Program., 11*, pp. 91–116, 1991.

[82] Z. Majkić, “Two-valued encapsulation of many-valued logic programming,” Technical Report, University ‘La Sapienza’, Roma, in http://www.dis.uniroma1.it/~majkic/, 2003.

[83] T. Przymusinski, “Every logic program has a natural stratification and an iterated fixed point model,” In *Eighth ACM Symposium on Principles of Databases Systems*, pp. 11–21, 1989.

[84] Z. Majkić, “Ontological encapsulation of many-valued logic,” In *19th Italian Symposium of Computational Logic (CILC04), June 16-17, Parma, Italy*, 2004.

[85] Z. Majkić, “Reduction of many-valued logic programs into two-valued modal logics,” In *Intellectual Archive Bulletin, Shiny World Corp., Toronto, ID:493, ISSN 1929-1329 (arXiv:1103.0920), 21 June*, pp. 1–27, 2012.

[86] Z. Majkić, “Many-valued intuitionistic implication and inference closure in a bilattice based logic,” In *35th International Symposium on Multiple-Valued Logic (ISMVL 2005), May 18-21, Calgary, Canada*, 2005.

[87] Z. Majkić, “Many-valued logic programming and fixpoint semantics for higher-order Herbrand models,” In *20th Workshop on (Constraint) Logic Programming (WLP 2006), February 22-24, Vienna, Austria*, 2006.

[88] A. Urquhart, “Semantics for relevant logics,” *J. Symb. Log., 37(1)*, pp. 159–169, 1972.

[89] R. Routley and R. K. Meyer, “Semantics of entailment I,” In *H. Leblanc (Ed.), Truth Syntax and Modality, Amsterdam: North-Holland*, pp. 199–243, 1973.

[90] K. Došen, “A brief survey of frames for the Lambek calculus,” *Z. Math. Log. Grundl. Math., 38*, pp. 179–187, 1992.

[91] J. M. Dunn, “Partial gaggles applied to logics with restricted structural rules,” In *P. Schröeder-Heister and K. Došen (Eds.), Substructural Logics, Logic and Computation 2, Oxford University Press*, pp. 63–108, 1993.

[92] M. Lenzerini, “Data integration: a theoretical perspective,” In *PODS ’02: Proceedings of the Twenty-First ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pp. 233–246, 2002.

[93] G. M. Kelly and A. J. Power, “Adjunctions whose counits are coequalizers, and presentations of finitary enriched monads,” *J. Pure Appl. Algebra, 89*, pp. 163–179, 1993.

[94] E. Moggi, “Notions of computation and monads,” *Inf. Comput., 93(1)*, pp. 55–92, 1991.

[95] E. Moggi, “Computational lambda-calculus and monads,” In *Proceedings of the Fourth Annual Symposium on Logic in computer science*, pp. 14–23, 1989.

[96] P. Buneman, S. Naqui, V. Tanen, and L. Wong, “Principles of programming with complex objects and collection types,” *Theor. Comput. Sci., 149(1)*, 1995.

[97] G. D. Plotkin and A. J. Power, “Adequacy for algebraic effects,” In *Proc. FOSSACS 2001, LNCS 2030*, pp. 1–24, 2001.

[98] A. Calì, D. Calvanese, G. De Giacomo, and M. Lenzerini, “Data integration under integrity constraints,” In *AISE 2002*, pp. 262–279, 2002.

[99] R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa, “DATA exchange: semantics and query answering,” In *Database Theory — ICDT 200*, pp. 207–224, 2003.

[100] A. Y. Halevy, “Theory of answering queries using views,” *SIGMOD Rec., 294)*, pp. 40–47, 2000.

[101] J. Baez and J. Dolan, “Higher-dimensional algebra III: n-categories and the algebra of opetopes,” *Adv. Math., 135*, pp. 145–206, 1998.

[102] P. M. Cohn, “Universal Algebra,” *London: Harper and Row*, 1965.

[103] M. Smith and G. Plotkin, “The category-theoretic solution of recursive domain equations,” *SIAM J. Comput.*, 1982.

[104] J. Adamek and J. Rosicky, “Locally Presentable and Accessible Categories,” *London Mathematical Society Lecture Note Series 189, Cambridge University Press*, 1994.

[105] G. M. Kelly, “Basic Concepts of Enriched Category Theory,” *London Mathematical Society Lecture Note Series 64, Cambridge University Press*, 1982.

[106] P. Gabriel and F. Ulmer, “Lokal prasentierbare kategorien,” *Lecture Notes in Mathematics 221, Berlin: Springer*, 1971.
[107] A. J. Power, “Enriched Lawvere theories,” In *Theory and Applications of Categories*, pp. 83–93, 2000.
[108] Y. Kinoshita, A. J. Power, and M. Takeyama, “Sketches,” *J. Pure Appl. Algebra, 143*, pp. 275–291, 1999.
[109] M. Barr and C. Wells, “Toposes, Triples and Theories,” *Grundelehren der math. Wissenschaften 278, Springer*, 1985.
[110] C. Lair, “Sur le genre d’esquissibilite des categories modelables (accessibles) possedant les produits de deux,” *Diagrammes, 35*, pp. 25–52, 1996.
[111] M. Makkai and R. Pare, “Accessible Categories: The Foundations of Categorical Model Theory,” *Contemporary Mathematics 104, Amer. Math. Soc.*, 1989.
[112] Z. Majkić, “DB category: denotational semantics for view-based database mappings,” *arXiv:1103.0217v1, 24 February*, pp. 1–40, 2011.
[113] H. Rasiowa and R. Sikorski, “The Mathematics of Metamathematics,” *Warsaw: PWN- Polisch Scientific Publishers, 3rd edition*, 1970.
[114] G. Gentzen, “Über die Existenz unabhängiger Axiomensysteme zu unendlichen Satzsystemen,” *Math. Ann., 107*, pp. 329–350, 1932.
[115] P. Hertz, “Über Axiomensysteme für beliebige Satzsysteme,” *Math. Ann., 101*, pp. 457–514, 1929.
[116] P. Blackburn, J. F. Benthem, and F. Wolter, “Handbook of Modal Logic,” *Studies in Logic and Practical Reasoning 3, Elsevier Science Inc.*, 2006.
[117] M. Fréchet, “Abstract sets, abstract spaces and general analysis,” *Math. Mag., 24*, pp. 147–155, 1951.
[118] W. Lawvere, “An elementary theory of the category of sets,” *Proc. Natl. Acad. Sci. USA, 52*, pp. 1506–1511, 1964.
[119] M. Makkai, “Towards a categorical foundation of mathematics,” In *Logic Colloquium 95 (Haifa), Lecture Notes Logic. 11, Berlin, Germany: Springer*, pp. 153–190, 1998.
[120] W. Lawvere, “Variable quantities and variable structures in topoi,” In *Algebra, Topology, and Category Theory (A Collection of Papers in Honor of Samuel Eilenberg), New York: Academic Press*, pp. 101–131, 1976.
[121] A. Y. Halevy, Z. G. Ives, J. Madhavan, P. Mork, D. Suciu, and I. Tatarinov, “The Piazza peer data management system,” *IEEE Trans. Knowl. Data Eng., 16(7)*, pp. 787–798, 2004.
[122] A. Fuxman, P. G. Kolaitis, R. Miller, and W. C. Tan, “Peer data exchange,” *ACM TODS, 31(4)*, pp. 1454–1498, 2006.
[123] M. A. V. Salles, J. P. Dittrich, S. K. Karakashian, O. R. Girard, and L. Blunschi, “iTrial: pay-as-you-go information integration in dataspaces,” In *Proc. of VLDB, Vienna, Austria*, pp. 663–674, 2007.
[124] P. Bernstein, A. Halevy, and R. Pottinger, “A vision for management of complex models,” *SIGMOD Rec., 29(4)*, pp. 55–63, 2000.
[125] G. Rosolini, “Continuity and effectiveness in Topoi,” *Ph.D dissertation, Oxford University, Also preprint, Department of Computer Science, Carnegie-Mellon University*, 1986.
[126] E. F. Codd, “A relational model of data for large shared data banks,” *Commun. ACM, 13(6)*, pp. 377–387, 1970.
[127] S. Abiteboul, R. Hull, and V. Vianu, “Foundations of Databases,” *Reading, Massachussetts: Addison Wesley Publ. Co.*, 1995.
[128] Z. Diskin, “Generalized sketches as an algebraic graph-based framework for semantic modeling and database design,” *Laboratory for Database Design, FIS/LDBD-97-03, May*, 1997.
[129] M. Makkai, “Generalized sketches as a framework for completeness theorems,” *Technical report, McGill University*, 1994.
[130] Z. Majkić, “Algebraic operators for matching and merging of relational databases,” In *International Conference in Artificial Intelligence and Pattern Recognition (AIPR-09), Orlando FL, USA, July 13–16*, 2009.
[131] Z. Majkić, “Matching, merging and structural properties of data base category,” *arXiv:1102.2395v1, 11 February* pp. 1–27, 2011.
[132] P. Buneman, S. Davidson, and A. Kosky, “Theoretical aspects of schema merging,” In *EDBT 1992*, pp. 152–167, 1992.

[133] A. K. Chandra and D. Harel, “Structure and complexity of relational queries,” *J. Comput. Syst. Sci., 25(1)*, pp. 99–128, 1982.

[134] D. McLeod and D. Heimbigner, “A federated architecture for information management,” *ACM Trans. Inf. Syst., 3(3)*, pp. 253–278, 1985.

[135] A. Sheth and J. Larsen, “Federated database systems for managing distributed, heterogeneous and autonomous databases,” *ACM Comput. Surv., 22(3)*, pp. 183–236, 1990.

[136] A. Calì, D. Calvanese, G. De Giacomo, and M. Lenzerini, “Reasoning in data integration systems: why LAV and GAV are siblings,” In *Proceedings of the 14th International Symposium on Methodologies for Intelligent Systems, ISMIS 2003, 2871, Springer*, pp. 282–289, 2003.

[137] R. Fagin, “Inverting schema mappings,” *ACM TODS, 32(4)*, pp. 25–78, 2005.

[138] E. Franconi, G. Kuper, A. Lopatenko, and L. Serafini, “A robust logical and computational characterization of peer-to-peer data systems,” *Technical Report DIT-03-051, University of Trento, Italy, September*, 2003.

[139] A. Kementsietsidis, M. Arenas, and R. J. Miller, “Mapping data in peer-to-peer systems: semantics and algorithmic issues,” In *Proc. of SIGMOD*, 2003.

[140] Z. Majkić, “Weakly-coupled ontology integration of P2P database systems,” In *1st Int. Workshop on Peer-to-Peer Knowledge Management (P2PKM), August 22, Boston, USA*, 2004.

[141] Z. Majkić, “Weakly-coupled P2P system with a network repository,” In *6th Workshop on Distributed Data and Structures (WDAS’04), July 5-7, Lausanne, Switzerland*, 2004.

[142] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati, “Inconsistency tollerance in P2P data integration: an epistemic approach,” In *Proc. 10th Int. Workshop on Database Programming Language*, 2005.

[143] S. Davidson, P. Buneman, and A. Kosky, “Semantics of database transformations,” In *B. Thalheim and L. Libkin (Eds.), Semantics in Databases, LNCS, 1358*, pp. 55–91, 1998.

[144] R. Miller, L. Haas, and M. Hernandez, “Schema mapping as query discovery,” In *Proc.VLDB*, pp. 77–88, 2000.

[145] A. Y. Halevy, D. Suiu, Z. G. Ives, and I. Tatarinov, “Schema mediation for large-scale semantic data sharing,” *VLDB J., 14(1)*, pp. 68–83, 2005.

[146] J. Madhavan, P. A. Bernstein, P. Domingos, and A. Y. Halevy, “Representing and reasoning about mappings between domain models,” In *AAAI/IAAI*, pp. 80–86, 2002.

[147] M. Lenzerini and Z. Majkić, “General framework for query reformulation,” *Semantic Webs and Agents in Integrated Economies, D3.1, IST-2001-34825, February*, 2003.

[148] M. Lenzerini and Z. Majkić, “First release of the system prototype for query management,” *Semantic Webs and Agents in Integrated Economies, D3.3, IST-2001-34825*, 2003.

[149] D. Beneventano, M. Lenzerini, F. Mandreoli, and Z. Majkić, “Techniques for query reformulation, query merging, and information reconciliation-part A,” *Semantic Webs and Agents in Integrated Economies, D3.2.A, IST-2001-34825*, 2003.

[150] I. Zaihrayeu, “Query answering in peer-to-peer database networks,” *Technical Report DIT-03-012, University of Trento, Italy*, 2003.

[151] Z. Majkić, “Flexible intentional query-answering for RDF peer-to-peer systems,” In *7th International Conference on Flexible Query Answering Systems (FQAS 2006), 7–10 June, Milano, Italy*, 2006.

[152] Z. Majkić and B. Prasad, “Soft query-answering computing in P2P systems with epistemically independent peers,” In *Book on Soft Computing Applications in Industry, STUDFUZZ 226, Berlin: Springer*, pp. 331–356, 2008.

[153] R. Fagin, “Horn clauses and database dependencies,” *J. ACM, 29*, pp. 952–985, 1982.

[154] J. Madhavn and A. Halevy, “Composing mappings among data sources,” In *Proc. of VLDB*, pp. 572–583, 2003.

[155] R. Fagin, P. G. Kolaitis, L. Popa, and W. Tan, “Composing schema mappings: second-order dependencies to the rescue,” *ACM TODS, 30(4)*, pp. 994–1055, 2005.

[156] S. Melnik, P. A. Bernstein, A. Halevy, and E. Rahm, “Supporting executable mapping in model management,” In *SIGMOD*, pp. 167–178, 2005.
[157] A. Nash, P. Bernstein, and S. Melnik, “Composition of mappings given by embedded dependencies,” In *Proc. of PODS*, pp. 172–183, 2005.
[158] R. Fagin, P. G. Kolaitis, L. Popa, and W. Tan, “Quasi-inverses of schema mappings,” *ACM TODS, 33(2)*, pp. 1–52, 2008.
[159] A. Corradini, “A complete calculus for equational deduction in coalgebraic specification,” *Report SEN-R9723, National Research Institute for Mathematics and Computer Science, Amsterdam*, 1997.
[160] J. F. Adams, “Infinite loop spaces,” *Princeton: Princeton U. Press*, 1978.
[161] J. M. Boardman and R. M. Vogt, “Homotopy Invariant Structures on Topological Spaces,” *Lecture Notes in Mathematics 347, Berlin: Springer*, 1973.
[162] J. P. May, “Simplicial Objects in Algebraic Topology,” *Princeton: Van Nostrand*, 1968.
[163] J. C. Baez and J. Dofan, “Categorification,” In *E. Getzler and M. Kapranov (Eds.), Workshop on Higher Category Theory and Physics, March 28-30, Northwestern University, Evanston, IL*, 1997.
[164] C. Ehresmann, “Introduction to the theory of structured categories,” *Technical Report 10, University of Kansas*, 1966.
[165] M. Barr and C. Wells, “The formal description of data types using sketches,” In *Mathematical Foundations of Programming Language Semantics, LNCS 298, Springer*, 1988.
[166] P. J. Freyd and G. M. Kelly, “Categories of continuous functors, I,” *J. Pure Appl. Algebra, 2*, pp. 169–191, 1972.
[167] D. I. Spivak, “Kleisli database instances,” *arXiv:1209.1011, 05 Sep*, pp. 1–23, 2012.
[168] R. Penrose and W. Rindler, “Spinors and Space-Time, vol. 1,” *Cambridge: Cambridge University Press*, 1984.
[169] F. W. Warner, “Foundations of Differentiable Manifolds and Lie Groups,” *New York: Springer*, 1983.
[170] A. Kosinski, “Differentiable Manifolds,” *San Diego, CA: Academic Press, Inc.*, 1992.
[171] G. F. Torres’delCastillo, “Differentiable Manifolds-A Theoretical Physics Approach,” *New York, NY, USA: Springer*, 2010.

Index

abstract categorial symmetries 65
Abstract Object Types (AOT) 350
adjunctions 21, 45, 344
adjunctions-as-fields 41, 58, 62
algebraic lattice 281, 305, 306, 323
arrow categories, *n*-dimensional levels 1, 6, 29
autoreferential Kripke semantics 208, 216

Belnap's 4-valued epistemic logic 209
bicategories and 2-categories 178
bisimulation for coalgebras 202
bisimulation relation 176

Calculus of Nondeterministic Mobile Processes 174
Cartesian Closed Category (CCC) 126–128, 147, 149, 151, 156, 157, 166, 169, 280, 347, 349
category of idempotents $\mathbf{ER}_b$ 155, 157, 166
Category **Z** of Finite Labelled Trees 183
Christoffel symbols 390
Classic Propositional Logic (CPL) 326, 333
closure operator 288, 289, 302, 305, 306, 308, 325, 329
coalgebraic semantics 350
coinductive iteration 204
(co)limits 40, 58, 345
comma lifting operator 6, 9
comma projections 1
comma-induction transformation 16, 18, 21
comma-propagation transformation 29, 34, 36, 38, 41, 58
comonad 251, 282, 347
complete lattice 263, 302, 304–306, 314, 323
Compton wavelength 384
conceptual categorial symmetry 72, 78, 84, 113, 152, 185, 191, 201, 248, 266, 271, 278, 288, 290
conceptually closed categories (CoCC) 84
congruence relation 326
covariant implication functor 124, 126

database category **DB** 221
Database management system (DBMS) 257, 263, 295, 352, 376
denotational semantics 183, 252, 282, 287, 346, 360, 369
determinism 383
differentiable smooth manifolds 388

embedded implication dependencies (EID) 354
encapsulation operator J 2
equality generating dependency (egd) 354, 357
Euler–Lagrange equation 385

finite labelled trees (FTS) 182
First-Order Logic (FOL) 329, 353, 355, 362
fixpoint operators 148, 156, 166
formulae-as-types 123, 144

Galilean boost 69, 82
Galois connection 344
Gentzen–Prawitz natural deduction 111
global symmetries 58
global-and-local-as-view (GLAV) 222, 348, 353, 369
group unitary $U(1)$ 71

Hennessy–Milner logic 202
Heyting algebra 324
hidden action 173
homomorphism of algebras 298, 299, 301, 306, 307, 325, 352, 361

imploded categories 107
information flux 220, 224, 226, 227, 231, 233, 241, 246, 254, 287, 290, 292, 295, 312, 373, 374, 376, 379, 380
initial algebra 282, 298, 299, 301, 308, 326
integrity constraints 352, 354, 357, 375
intensional abstraction operator 73
Internal symmetries 71
intuitionistic logic **IC** category 143
Intuitionistic Logic (IL) 112, 123, 138–140, 210, 327
invariant Noether's charge 387
inverse of encapsulation operator $\psi = J^{-1}$ 3

Jacobian coord.transformation 392

K-theory 170
Karoubi envelope 153, 170–172
Kleisli category 283, 285, 351, 370
Knaster–Tarski fixpoints 323
Kripke models 201, 202, 218, 333

labelled transition system (LTS) 175
lambda calculus 144
light-cone 381
Lindenbaum–Tarski algebra 358
LJ sequent system 139

https://doi.org/10.1515/9783111081670-010

many-valued logics 208
mapping-interpretations 221, 232, 368, 372, 373, 376, 380
Minkowski metric 381
Minkowski time-space 383
monad 251, 282, 283, 287, 298, 346, 369
monadic algebras 253, 275, 282, 284
monoid A^* of finite sequences of actions 177
monoidal category 292, 295, 313, 315, 347

Natural Deduction category **ND** 120
Noether theorem 57

ontological encapsulation 212, 214, 215
operads 220, 363, 373
operational semantics 174

partitioned adjunction (p-adjunction) 129
peer-to-peer (P2P) 348, 353
perfectly symmetric categories 80
power-view operator 249–251, 288, 312, 320, 350, 351, 358, 360
pullbacks in **Rel** category 194, 198, 203

quotient algebra 326

reflexive objects 149, 150
reification data 74
Relational Database (RDB) 352
Ricci scalar 61

Scott domains 151
Second Order tgd (SOtgd) 251, 356
Select-Project-Rename-Join-Union (SPRJU) algebra 358
semantic-reflection transformation 213
sequent-based natural deduction 116
signature of algebra 298, 300, 325, 362
sketch category 222, 245, 367, 376, 378, 380
strong bisimulations as fixpoints 205
symmetry-extended categories 95

tangent space of vectors T_pM 390
Tarski's interpretations 332, 355, 368
transition systems 173
transition systems as fixpoints 198
tuple generating dependency (tgd) 348, 353, 354, 356, 380

variety 299
vector field 392

Weak monoidal Topos 351, 370
worldline 383