31268 Web Systems

UNIVERSITY OF TECHNOLOGY SYDNEY
Faculty of Information Technology

Week 08: Computer Science 2
Part 1: Intro to Logic

Mathematics and Computer Science

UNIVERSITY OF TECHNOLOGY SYDNEY
Faculty of Information Technology

- **Under the bonnet**
  - Representation of information
  - Number systems
- **Logic and Mathematics**
  - Boolean Algebra
  - Binary Arithmetic
- **Storage and processing of information**
  - Computation
  - Memory
  - Coding

Objectives and Motivation

You must learn to:

- Write logical expressions
- Understand when two logical expressions mean the same thing

In order to:

- Translate between your reasoning and the computer's reasoning.
- Find the best way to write code and make chips!

Boolean Algebra

- Also called **Boolean Logic**
- Outlined in 1854, by an English mathematician, George Boole
- A system that contains only 2 values, conventionally labeled TRUE and FALSE

We all know about logic

- (This is a lecture) OR (This year is 1942)
- (This is a lecture) AND (This year is 1942)
- (This year is 1942)
- NOT(This year is 1942)

We all know about logic

- (This is a lecture) OR (This year is 1942) ✓ **True**
- (This is a lecture) AND (This year is 1942)
- (This year is 1942)
- NOT(This year is 1942)

We all know about logic

- (This is a lecture) OR (This year is 1942) ✓ **True**
- (This is a lecture) AND (This year is 1942) ✗ **False**
- (This year is 1942)
- NOT(This year is 1942)

We all know about logic

- (This is a lecture) OR (This year is 1942) ✓ **True**
- (This is a lecture) AND (This year is 1942) ✗ **False**
- (This year is 1942) ✗ **False**
- NOT(This year is 1942)

We all know about logic

- (This is a lecture) OR (This year is 1942) ✓ **True**
- (This is a lecture) AND (This year is 1942) ✗ **False**
- (This year is 1942) ✗ **False**
- NOT(This year is 1942) ✓ **True**

We all know about logic

- (This is a lecture) OR (This year is 1942) ✓ **True**
- (This is a lecture) AND (This year is 1942) ✗ **False**
- (This year is 1942) ✗ **False**
- NOT(This year is 1942) ✓ **True**

http://handbook.uts.edu.au/subjects/31272.html

Basic Operations

Basic Operations

- Three Basic Operations:

  AND, OR and NOT

- Boolean operations also called Boolean Functions

- We will also look at the Boolean XOR operation – it's relevant to Binary Addition

Boolean Operations

- Every Boolean operation has inputs and outputs
- NOT operation has 1 input and 1 output
- AND and OR operations have 2 or more inputs and 1 output

Formalising the AND Operation

- A **TRUTH TABLE** shows all combinations of inputs and resulting output.
- Boolean TRUTH TABLE usually uses TRUE and FALSE
  - 0 corresponds to FALSE, 1 corresponds to TRUE
- Difference between the two is only the notation used

Truth table for AND Operation

Inputs: A	B	**Output:** A **and** B
False	False	False
True	False	False
False	True	False
True	True	True

The output only becomes "TRUE" when **BOTH** inputs are "TRUE"

Substituting 0 and 1

Inputs: **A**	**B**	**Output:** **A and B**
0	0	0
1	0	0
0	1	0
1	1	1

- You can substitute 0 for "FALSE" and 1 for "TRUE"

TRUTH TABLE for OR Operation

Inputs: A	B	**Output:** A **or** B
False	False	False
True	False	True
False	True	True
True	True	True

The OR Operation outputs TRUE when **Either or both** of its Inputs is TRUE

Truth table for NOT Operation

Inputs: A	**Output:** not A
False	True
True	False

- The NOT Operation outputs the **opposite** of its Input

Truth table for XOR Operation

Inputs: A	B	**Output:** A xor B
False	False	False
True	False	True
False	True	True
True	True	False

- e**X**clusive **OR**
- More like the "English OR statement"
  - e.g. "Either red or black"
- Equal to the operation
  - (A OR B) AND NOT(A AND B)

Computer Languages

- Programs are full of Boolean Operations!
- In the programming language Java there are Boolean Variables

- Boolean Variables in Java can only be TRUE or FALSE

Boolean Operators in Java

- The **&&** operator is an AND Operator in Java
  - In Java the statement (a && b) has value **true** only when a and b are both **true**
- The **||** operator is an OR Operator in Java
  - In Java the statement (a || b) has value **true** when either a or b (or both) are **true**
- The **!** Operator is a NOT Operator in Java
  - The statement (!a) has the value **true** only when a is **false**

Example:

- Below is a fragment of Java Code. a and y are integer variables. p and q are boolean variables. The symbols > and < stand for "Greater than" and "Less than" respectively.

```
int x = 1;
int y = 2;
p = ( x > 1 ) || ( y > 1 );
q = ( x > 4 ) && ( y < 5 );
```

What are the values of p and q ?
(Hint: They can only be true or false)

Take a break: sloppy writing

Algebraists write nothing for *product* and just + for *plus*, logicians can be lazy too…

- Algebra: x times y plus z = xy + z
- Logic: (not(x) and y) or z = Xy + z

What sloppy writing is:

- **CAPITALS** denote not: X = not(x)
- + denotes or: X + y = not(x) or y
- Nothing is and and and is nothing·:

Xy = not(x) and y

- and has greater priority than or:

Xy + z = (not(x) and y) or z

… and what it isn't

Note: This is **not arithmetic,** so when we're writing using logic

1111 + 0001 = 1 + 0 = TRUE OR FALSE = 1

not 10000 as in arithmetic.

Other representations

- AND (formally called a **conjunction**)
  - x ∧ y
  - x . Y
- OR (formally called a **disjunction**)
  - x ∨ y
  - x + y
- NOT (formally called a **negation** or **complement**)
  - ¬ x
  - ! x

Questions?