# The Acceptance Problem - Undecidable Languages

Lecture 29
Section 4.2

Robb T. Koether

Hampden-Sydney College

Fri, Nov 7, 2014

# Outline

# Assignment

## Assignment

- Chapter 5: Exercises 5, 6, 7.
- Chapter 5: Problems 15, 16, 19, 24, 25, 28.

# Outline

# Universal Turing Machines

**Definition (Universal Turing machine)**

A universal Turing machine is a Turing machine that can read a description of any Turing machine and simulate it on any input.

- Do universal Turing machines really exist?

# Universal Turing Machines

- Yes. We call them programmable computers.
- They read a description of a Turing machine, which we call a program.
- Then they simulate the Turing machine, which we call executing the program.
- This should all sound familiar.

# Outline

1. Assignment
2. Universal Turing Machines
3. **The Acceptance Problem for Turing Machines**
4. Turing-Unrecognizable Languages

# The Acceptance Problem for Turing Machines

## The Acceptance Problem for Turing Machines

Given a Turing machine $M$ and a string $w$, does $M$ accept $w$?

- The language is

$$A_{\text{TM}} = \{\langle M, w\rangle \mid M \text{ accepts } w\}.$$

# ATM is Turing-Recognizable

**Theorem**

*$A_{TM}$ is Turing-recognizable.*

**Proof.**

- Build a universal Turing machine *U* and use it to simulate *M* on the input *w*.
- If *M* accepts *w*, then *U* will halt in its accept state.
- If *M* does not accept *w*, then *U* may halt in its reject state or it may loop.
- That is why *U* is only a recognizer, not a decider.

□

# Outline

**Theorem**

*There exist Turing-unrecognizable languages.*

- We will establish this by showing that the function

$$f : \mathcal{M} \to \mathcal{L}$$
$$f : M \mapsto L(M)$$

from the set $\mathcal{M}$ of all Turing machines to the set $\mathcal{L}$ of all languages is not onto.

**Lemma**

*The set $\mathcal{M}$ of all Turing machines is a countably infinite set.*

# Turing-Unrecognizable Languages

## Proof of the lemma.

- Each Turing machine has a finite description:

$$(Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}}).$$

- Express the description in binary, perhaps by using ASCII.
- Thus, the set of all Turing machines is represented by an infinite set of finite binary strings.
- We already know that every infinite set of finite binary strings is countable.
- (They can be arranged in lexicographical order.)

□

**Lemma**

*The set $\mathcal{B}$ of all infinite binary strings is uncountable.*

- To prove this, we need to use a diagonalization argument, which is based on proof by contradiction.

## Proof of the lemma.

- Suppose $\mathcal{B}$ is countable.
- Then its members can be listed $w_1, w_2, w_3, \ldots$.
- (Each $w_i$ is an infinitely long binary string.)
- Create a binary array that is infinite to the right and down by letting row $i$ be the bits in $w_i$, for $i = 1, 2, 3, \ldots$.

$\square$

## Proof of the lemma.

- For example, we might have

$$\begin{array}{lcccccccccc}
w_1 & = & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & \ldots \\
w_2 & = & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & \ldots \\
w_3 & = & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & \ldots \\
w_4 & = & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & \ldots \\
w_5 & = & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & \ldots \\
w_6 & = & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & \ldots \\
w_7 & = & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & \ldots \\
w_8 & = & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & \ldots \\
 & \vdots & & & & & & & & &
\end{array}$$

$\square$

# Turing-Unrecognizable Languages

## Proof of the lemma.

- Define an infinite binary string $w$ as follows.
  - If the $i^{\text{th}}$ bit in row $i$ is 0, set the $i^{\text{th}}$ bit of $w$ to 1.
  - If the $i^{\text{th}}$ bit in row $i$ is 1, set the $i^{\text{th}}$ bit of $w$ to 0.

$\square$

# Turing-Unrecognizable Languages

## Proof of the lemma.

$$
\begin{array}{lcccccccccc}
w_1 & = & \color{red}{0} & 1 & 1 & 0 & 0 & 1 & 1 & 1 & \ldots \\
w_2 & = & 1 & \color{red}{1} & 0 & 0 & 0 & 1 & 1 & 1 & \ldots \\
w_3 & = & 1 & 1 & \color{red}{0} & 1 & 0 & 0 & 0 & 0 & \ldots \\
w_4 & = & 0 & 0 & 1 & \color{red}{0} & 0 & 1 & 0 & 1 & \ldots \\
w_5 & = & 0 & 1 & 0 & 1 & \color{red}{0} & 1 & 1 & 0 & \ldots \\
w_6 & = & 0 & 0 & 0 & 1 & 1 & \color{red}{0} & 1 & 1 & \ldots \\
w_7 & = & 1 & 0 & 0 & 1 & 0 & 0 & \color{red}{1} & 0 & \ldots \\
w_8 & = & 1 & 0 & 1 & 1 & 0 & 1 & 1 & \color{red}{0} & \ldots \\
\vdots & & & & & & & & & &
\end{array}
$$

- So $w = 10111101\ldots$

□

## Proof of the lemma.

- Certainly, $w$ is an infinite binary string.
- But $w$ cannot be in the list of all infinite binary strings because it disagrees with every $w_i$ in the list.
- This is a contradiction.
- Therefore, $\mathcal{B}$ must be uncountable.

□

# Turing-Unrecognizable Languages

## Proof of the theorem.

- It is easy to see that there is a one-to-one correspondence between $\mathcal{L}$ and $\mathcal{B}$.
- Define a function $g : \mathcal{L} \to \mathcal{B}$ as follows.
- For any language $L$, $g$ maps $L$ to the infinite binary string $s$ defined by the following procedure.
- List of all the finite binary strings and order the list lexicographically and number the strings $w_1, w_2, w_3, \ldots$

$\square$

# Turing-Unrecognizable Languages

## Proof of the theorem.

- For every $i \geq 1$,
  - If $w_i \in L$, then the $i^{\text{th}}$ bit of $s$ is 1.
  - If $w_i \notin L$, then the $i^{\text{th}}$ bit of $s$ is 0.
- Clearly, this is a one-to-one correspondence between the set of all languages and the set of all infinite binary strings.
- Thus, the set of all languages is uncountable.

$\square$

# Turing-Unrecognizable Languages

## Proof of the theorem.

- Now suppose that every language is recognizable.
- Then the function $f : \mathcal{M} \to \mathcal{L}$ defined by $f : M \mapsto L(M)$ is onto.
- That implies that $|\mathcal{M}| \geq |\mathcal{L}|$, which is impossible since $\mathcal{M}$ is countable and $\mathcal{L}$ is uncountable.
- Thus, there exist languages that are not Turing-recognizable.

$\square$