Theory of Computation – Lecture 9

MARCH 1, 2021

- Undecidability (Section 4.2)
- Undecidable Problems from Language Theory (Section 5.1)
- Mapping Reducibility (Section 5.3)
Announcements

- Exam 1 is graded.
- Grades will be released after class. The curve will be described in the email from Gradescope.
- Regrades/grade questions can be submitted via Gradescope or in office hours for one week.
Quick Recap

- Last lecture, we studied decidability.
- We proved that several problems are decidable by creating a TM decider for the language of the problem.
  - Does a given DFA accept an input string \( w \)?
  - Does a given NFA accept an input string \( w \)?
  - Is the language of a given DFA empty?
  - Do two DFAs recognize the same language?
  - Is the language of a given CFG empty?
- Then, we moved on to showing that some problems are undecidable.
  - There are languages that are not decided by any Turing Machine – still TODO.
Quick Recap

- **Theorem:** If \( S \) is an infinite countable set, then the power set of \( S \) is uncountable.

- Let \( \Sigma \) be an alphabet. The set of all strings over \( S = \Sigma^* \) is infinite and countable.
  - The set of all strings over an alphabet is infinite and countable.
  - A language is a subset of \( \Sigma^* \).

- The power set of \( S \) is the set of all languages over \( \Sigma \).
  - **Corollary:** The set of all languages is uncountable.

- **Theorem:** Some languages are not Turing-recognizable.
  1. The set of Turing Machines is countable – it is a subset of a set of all strings.
  2. The set of languages is uncountable – it is the power set of a set of all strings.

**Thus – there are more languages than Turing Machines.**
Does a given Turing Machine \( M \) accept an input string \( w \)?

\[ A = \{ \langle M, w \rangle \mid M \text{ is a Turing machine and } M \text{ accepts } w \} \]

**Theorem:** \( A \) is Turing-recognizable.

**Proof:** Construct a TM \( U \) that recognizes \( A \)

\( U \): on input \( \langle M, w \rangle \) where \( M \) is a TM and \( w \) is a string

1. Simulate \( M \) on input \( w \).
2. If \( M \) enters accept state: output accept
   If \( M \) enters reject state: output reject
Does a given Turing Machine $M$ accept an input string $w$? $A = \{ (M, w) \mid M$ is a Turing machine and $M$ accepts $w \}$

**Theorem:** $A$ is undecidable.
Does a given Turing Machine $M$ accept an input string $w$?

$A = \{\langle M, w \rangle \mid M \text{ is a Turing machine and } M \text{ accepts } w\}$

**Theorem:** $A$ is undecidable.
Co-Turing Recognizability

- A language B is **co-Turing-recognizable** if it is the complement of a Turing-recognizable language.

- **Theorem**: A language is decidable if and only if it is Turing-recognizable and co-Turing-Recognizable.
A language $B$ is **co-Turing-recognizable** if it is the complement of a Turing-recognizable language.

**Theorem:** A language is decidable if and only if it is Turing-recognizable and co-Turing-Recognizable.
A Turing-Unrecognizable language

- Recall $A = \{\langle M, w \rangle | M$ is a Turing machine and $M$ accepts $w\}$
- **Theorem:** $\overline{A}$ is not Turing-recognizable.
Reducibility

CHAPTER 5
For two languages (problems) A and B, A is reducible to B if we can use a solution to B to solve A.

A: measure area of a rectangle  B: measure length of its sides

A: decide whether an NFA accepts a string w  B: decide whether a DFA accepts a string w
The Halting Problem

- Does a TM $M$ halt on input $w$?
- **Theorem:** $B = \{ (M, w) \mid M \text{ is a TM and } M \text{ halts on input } w \}$ is undecidable.
\[ A = \{ \langle M, w \rangle \mid M \text{ is a TM and } M \text{ accepts } w \} \]
\[ B = \{ \langle M, w \rangle \mid M \text{ is a TM and } M \text{ halts on input } w \} \]
The Empty Language Problem for Turing Machines

- \( E = \{(M, w) \mid M \text{ is a TM and } L(M) = \emptyset\} \)
- **Theorem**: \( E \) is undecidable.
\[ A = \{ \langle M, w \rangle \mid M \text{ is a TM and } M \text{ accepts } w \} \]

\[ E = \{ \langle M, \rangle \mid M \text{ is a TM and } L(M) = \emptyset \} \]
A function $f: \Sigma^* \rightarrow \Sigma^*$ is a **computable function** if some Turing machine $M$, on every input $w$, halts with just $f(w)$ on its tape.

Language $A$ is **mapping reducible** to language $B$ ($A \leq_m B$) if there is a computable function $f: \Sigma^* \rightarrow \Sigma^*$ (the **mapping reduction**) where $w \in A$ if and only if $f(w) \in B$. 

---

Mapping Reducibility