Lec 17. Decidable and undecidable languages

Eunjung Kim

Diagonal argument for uncountability of $[0, 1]$

0.8	1	3	4	2	0	$8 \cdots$
0.0	1	1	2	1	9	$0 \cdots$
0.2	0	3	1	4	1	$3 \cdots$
0.7	0	3	4	4	1	$3 \cdots$
0.1	0	2	7	4	9	$3 \cdots$
0.3	1	0	3	6	0	$1 \cdots$
0.2	4	3	1	4	7	$7 \cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

$\rightsquigarrow$ consider a real number $x = 0.\bar{8}\bar{1}\bar{3}\bar{4}\bar{4}\bar{0}\bar{7}\cdots = 0.7243186\cdots$
The perturbation on each digit can be arbitrary (just avoid using 0 and 9).

x is not listed by φ!

Diagonal argument for uncountability of $2^{\mathbb{N}}$

Diagonal argument: suppose φ lists all elements in $2^{\mathbb{N}}$.

- Rows and columns are indexed by $1, 2, \ldots$, i.e. $\mathbb{N}$
- i-th row corresponds to (the indicator vector of) the set $\varphi(i)$ in $2^{\mathbb{N}}$, with j-th entry being 1 if and only if j is in the set.
- Diagonalization step: construct a new set which is not listed by φ by flipping all the diagonal entries.

0	0	1	1	1	0	1 ···
0	1	1	1	1	1	0 ···
1	0	1	1	0	1	0 ···
1	0	1	0	1	1	0 ···
0	0	1	1	1	0	1 ···
0	1	0	1	1	0	1 ···
1	1	1	1	0	0	0 ···
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Consider the set

$$X = \overline{0}\,\overline{1}\,\overline{1}\,\overline{0}\,\overline{1}\,\overline{0}\,\overline{0}\cdots = 1001011\cdots$$

$\rightsquigarrow$ X is not listed by φ!

Countable or uncountable?

Use the fact that if there is a bijection between A and B, then there is a bijection between 2^A and 2^B.

- The collection of all languages over $\{0, 1\}$?
- $\{\tau(M) \in \{0, 1\}^* : M \text{ is a Turing machine}\}$ for a fixed encoding scheme τ?
- The collection of all languages over $\{0, 1\}$ recognizable by some Turing machine?

Language unrecognizable by TM

- The collection of all languages over $\{0, 1\}$? Uncountable.
- $\{\tau(M) \subseteq \{0, 1\}^* : M \text{ is a Turing machine}\}$? Countable.
- The collection of all languages over $\{0, 1\}$ recognizable by some Turing machine? Countable.

Language unrecognizable by TM

- The collection of all languages over $\{0, 1\}$? Uncountable.
- $\{\tau(M) \subseteq \{0, 1\}^* : M \text{ is a Turing machine}\}$? Countable.
- The collection of all languages over $\{0, 1\}$ recognizable by some Turing machine? Countable.

Unrecognizable

There is a language that cannot be recognized by any Turing machine.

Unrecognizable languages

Concrete example of unrecognizable language

Let $A_{TM} = \{(M, w) : \ M \text{ is a TM and } M \text{ accepts } w\}$.

Then $\bar{A}_{TM} := \{0, 1\}^* \setminus A_{TM}$ is not Turing-recognizable.

Follows from the undecidability of A_{TM} and the characterization of undecidable languages.

Decision Problem

Membership test for a language A

Consider a language $A \subseteq \Sigma^*$.

INPUT: a string $w \in \Sigma^*$.

TASK: decide if $w \in A$ or not; that is, output YES ("accept") if $w \in A$, output NO ("reject") otherwise.

The language A itself is also called a decision problem.

Solving a decision problem A

Solving a (decision) problem A means having an algorithm for A, i.e. an algorithm for the membership test for A. By Church-Turing Thesis, this means to have a Turing machine M which decides A, i.e.

$$M(w) = \begin{cases} \text{ACCEPT} & \text{if } w \in A \\ \text{REJECT} & \text{otherwise.} \end{cases}$$

Examples of Decision Problems

- Decide if a given context-free grammar G generates a given string w: corresponds to a membership test for the language

$$\{\langle G, w\rangle \mid G \text{ is a CFG that generates } w\}.$$

- Decide if a graph is connected: corresponds to a membership test for the language

$$\{\langle G\rangle \mid G \text{ is connected}\}.$$

- Shortest path problem, as a decision problem: corresponds to a membership test for the language

$$\{\langle G, s, t, L\rangle \mid \text{there is an } (s, t)\text{-path of length at most } L \text{ in } G\}.$$

- Halting problem, asking if a program (Turing machine) terminates on an input w, corresponds to a membership test for the langugge

$$\{\langle M, w\rangle \mid \text{a TM } M \text{ terminates on the input } w\ \}.$$

Solving a decision problem

For a language A i.e. a decision problem, A is

- decidable if there is an algorithm (= Turing machine) which decides A.
- undecidable if there is no Turing machine which decides A.

Examples of decidable languages

- $A_{DFA} = \{\langle B, w\rangle \mid$ B is a DFA that accepts input string w $\}$

Examples of decidable languages

- $A_{NFA} = \{\langle B, w\rangle \mid B \text{ is a NFA that accepts input string } w\}$

Examples of decidable languages

- $A_{REX} = \{\langle R, w \rangle \mid$
  R is a regular expression that generates input string w $\}$

Examples of decidable languages

- $E_{DFA} = \{\langle A \rangle \mid A \text{ is a DFA and } L(A) = \emptyset\}$

Examples of decidable languages

- $EQ_{DFA} = \{\langle A, B\rangle \mid A \text{ and } B \text{ are DFAs and } L(A) = L(B)\}$

Examples of decidable languages

- $A_{CFG} = \{\langle G, w\rangle \mid$ G is a CFG that generates input string w $\}$

Examples of decidable languages

- $A_{PDA} = \{\langle P, w\rangle \mid$
  P is a pushdown automaton that accepts input string w $\}$; caution

Examples of decidable languages

- Any context-free language A.

Inherently looping TM

First undecidable language

Consider the language $A_{TM} = \{\langle M, w\rangle \mid \quad \text{M is a TM and } M \text{ accepts } w.\}$.

1. We know that A_{TM} is Turing-recognizable; universal Turing machine.
2. But it is undecidable.

Inherently looping TM

1. Suppose that A_{TM} is decidable, i.e. there exists TM H such that

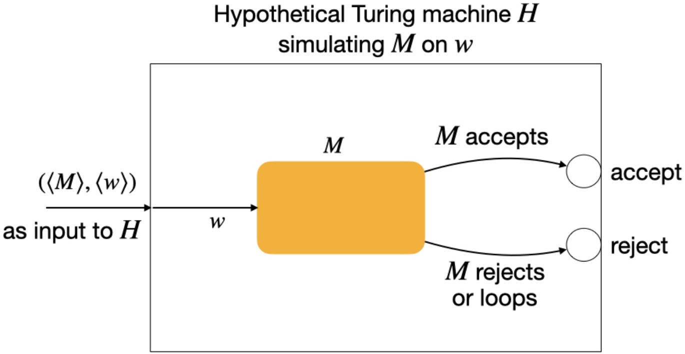
$$H(\langle M, w \rangle) = \begin{cases} \text{ACCEPT} & \text{if } M \text{ accepts } w \\ \text{REJECT} & \text{otherwise} \end{cases}$$

2. Consider a TM D gets a description $\langle M \rangle$ of an arbitrary TM M as input, and flips the answer of H on the input $\langle M, \langle M \rangle \rangle$, i.e.

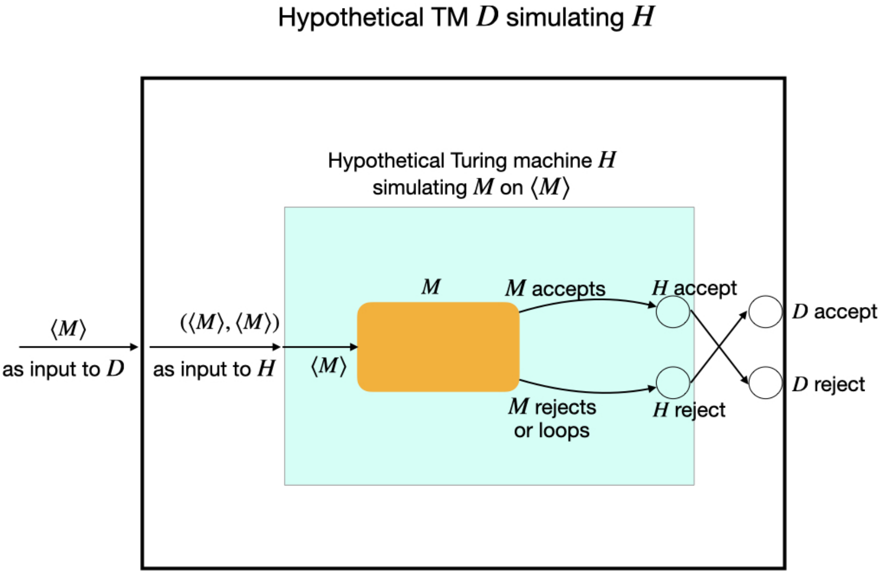
$$D(\langle M \rangle) = \begin{cases} \text{ACCEPT} & \text{if } H(\langle M, \langle M \rangle \rangle) = \text{REJECT} \\ \text{REJECT} & \text{if } H(\langle M, \langle M \rangle \rangle) = \text{ACCEPT} \end{cases}$$

3. What if we run TM D on the input $\langle D \rangle$, i.e the description of itself?

Inherently looping TM



INHERENTLY LOOPING TM



Inherently looping TM

Seen from the perspective of the diagonal argument.

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\cdots$
M_1	*accept*	*reject*	*accept*	*reject*	
M_2	*accept*	*accept*	*accept*	*accept*	$\cdots$
M_3	*reject*	*reject*	*reject*	*reject*	
M_4	*accept*	*accept*	*reject*	*reject*	
$\vdots$		$\vdots$			

Figure 4.20, Sipser 2012.

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\cdots$	$\langle D \rangle$	$\cdots$
M_1	*accept*	*reject*	*accept*	*reject*		*accept*	
M_2	*accept*	*accept*	*accept*	*accept*	$\cdots$	*accept*	$\cdots$
M_3	*reject*	*reject*	*reject*	*reject*		*reject*	
M_4	*accept*	*accept*	*reject*	*reject*		*accept*	
$\vdots$		$\vdots$			$\ddots$		
D	*reject*	*reject*	*accept*	*accept*		?	
$\vdots$		$\vdots$					$\ddots$

Figure 4.21, Sipser 2012.

Characterizing decidability

A language $A \subseteq \Sigma^*$ is said to be co-Turing-recognizable if its complement (i.e. $\Sigma^* \setminus A$) is Turing-recognizable.

Turing-recognizable and co-Turing-recognizable

A language is decidable if and only if it is Turing-recognizable and co-Turing-recognizable.

Characterizing decidability

A language $A \subseteq \Sigma^*$ is said to be co-Turing-recognizable if its complement (i.e. $\Sigma^* \setminus A$) is Turing-recognizable.

Turing-recognizable and co-Turing-recognizable

A language is decidable if and only if it is Turing-recognizable and co-Turing-recognizable.

- The direction ($\Rightarrow$) is straightforward.
- For the direction ($\Leftarrow$), let M_1 and M_2 be two TMs recognizing A and $\bar{A}$. Build a new TM M which runs both M_1 and M_2 simultaneously on $w \in \Sigma^*$ and outputs

$$M(w) = \begin{cases} \text{ACCEPT} & \text{if } M_1(w) = \text{ACCEPT} \\ \text{REJECT} & \text{if } M_2(w) = \text{ACCEPT} \end{cases}$$

  Clearly M decides A.