Theory of Formal Languages and Automata

Lecture 19

Mahdi Dolati

Sharif University of Technology

Fall 2023

December 1, 2023

Undecidability

- Are there tangible unsolvable problems?
  - Program verification:
    - A precise description,
    - A precise program,
    - Yet, we can not in general verify that the program performs as specified.

- We will see several computationally unsolvable problem.
  - Learn techniques to prove unsolvability.

Undecidability

- The acceptance problem for TMs:

$$A_{\mathsf{TM}} = \{\langle M, w\rangle |\ M \text{ is a TM and } M \text{ accepts } w\}.$$

Theorem
A_{TM} is undecidable.

- Observe that A_{TM} is Turing-recognizable:

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 ever enters its accept state, *accept*; if M ever enters its reject state, *reject*."

- If M loops on w, then U loops on $\langle M, w\rangle$:
  - U is not a decider.

Undecidability

- The acceptance problem for TMs:

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

Theorem
A_{TM} is undecidable.

- Observe that A_{TM} is Turing-recognizable:

$U =$ "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 ever enters its accept state, *accept*; if M ever enters its reject state, *reject*."

- If M loops on w, then U loops on $\langle M, w\rangle$:
  - U is not a decider.

U can not detect that M is not halting. We will see it is impossible.

Undecidability

- The acceptance problem for TMs:

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

Theorem
A_{TM} is undecidable.

- Observe that A_{TM} is Turing-recognizable:

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 ever enters its accept state, *accept*; if M ever enters its reject state, *reject*."

- U is an example of the **universal TM** (Turing'36).
- It is possible to **program** U.

Diagonalization Method

- We use diagonalization for the proof.

- A technique developed by Cantor 1873:
  - How to compare infinities? We can not count them.
  - We can compare them by trying to pair their elements.

Diagonalization Method

- A function f : A → B:
  - One-to-one (injective): Does not map different elements to the same place,
    $$a \neq b \rightarrow f(a) \neq f(b)$$
  - Onto (surjective): It hits every elements of B,
    $$\forall b \in B \ \exists a \in A \ f(a) = b$$
  - Correspondence (bijective): Both one-to-one and onto.

- A and B are the same size if there is f : A → B that is correspondence.

Diagonalization Method

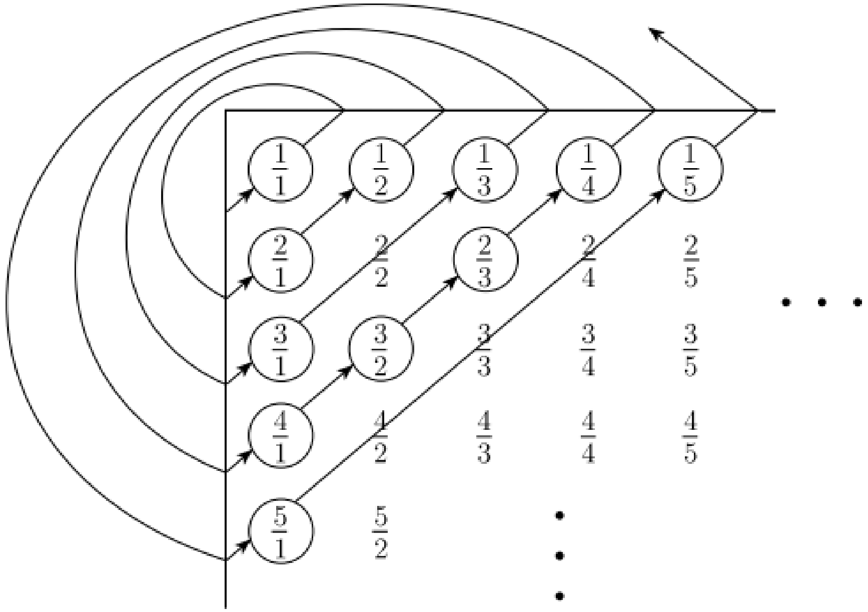
- Example: Consider the set of natural numbers (N) and even natural numbers (E).
- These sets have the same size, because there is correspondence f mapping N to E:

$$f(n) = 2n.$$

n	$f(n)$
1	2
2	4
3	6
$\vdots$	$\vdots$

Diagonalization Method

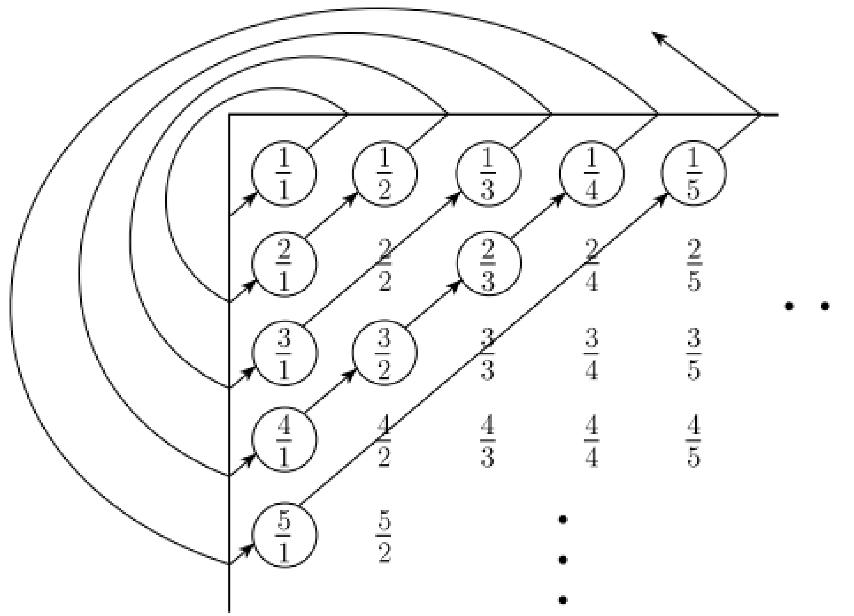
- **Definition**: A set A is countable if either it is finite or it has the same size as N.
- **Example**: Set of positive rational numbers, Q, is the same size as N.
  - We can turn the matrix into a list.



Diagonalization Method

- **Definition**: A set A is countable if either it is finite or it has the same size as N.
- **Example**: Set of positive rational numbers, Q, is the same size as N.
  - We can turn the matrix into a list.

If N and Q have the same size, do all infinities have the size?



Diagonalization Method

- There are sets that have no correspondence with N.
- These sets are very big.
- These sets are called **uncountable**.
- Cantor proved that the set of real numbers, R, is uncountable with a method called diagonalization.

Undecidability

Theorem

R is uncountable.

- **Proof**: We show that there is no correspondence between N and R. Suppose such correspondence f exists:
  - List all real numbers,
  - Construct a new real number that is not in the list,
    - Choose each digit to be different from one of the listed numbers,
  - Thus, the corresponding is not onto, a contradiction.
- Construct $0 \leq x \leq 1$ that is different from f(1) in the first decimal place and so on:
  - $X=0.464\ldots$

n	$f(n)$
1	3.14159...
2	55.55555...
3	0.12345...
4	0.50000...
⋮	⋮

Undecidability

- We can use the fact that R is uncountable as follows:

- Show that there are countably many TMs,
- Show that there are uncountably many languages,
- Each TM recognizes one language,
- Thus, there are languages that are not Turing-recognizable.

Undecidability

Corollary
Some languages are not Turing-recognizable.

- **Proof**:
  - Set of all strings on any alphabet is countable:
    - Write strings of length 0 first,
    - Then, write strings of length 1,
    - …
  - Set of all TMs is countable:
    - Each TM has an encoding (a string),
    - Omit strings that are not a TM,
    - The remaining strings are countable.

Undecidability

Corollary
Some languages are not Turing-recognizable.

- **Proof**:
  - Set of all strings on any alphabet is countable.
  - Set of all TMs is countable.
  - Set of all languages is uncountable:
    - Set of all infinite binary sequences, **B**, is uncountable:
      - Use diagonalization:

000000…
0**1**0100…
10**0**100… → **1011…**
111**0**00…

Undecidability

Corollary

Some languages are not Turing-recognizable.

- **Proof**:
  - Set of all strings on any alphabet is countable.
  - Set of all TMs is countable.
  - Set of all languages, **L**, is uncountable:
    - Set of all infinite binary sequences, **B**, is uncountable.
    - There is a correspondence between **L** and **B**:
      - Each language is a selecting from the set of all strings,
      - Selection is a binary decision,
      - Thus, the set of all languages is uncountable.

$$
\begin{array}{rcccccccccccl}
\Sigma^* = \{ & \varepsilon, & 0, & 1, & 00, & 01, & 10, & 11, & 000, & 001, & \cdots & \} & ; \\
A = \{ & & 0, & & 00, & 01, & & & 000, & 001, & \cdots & \} & ; \\
\chi_A = & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & \cdots & & .
\end{array}
$$

characteristic sequence of A → χ_A

Undecidability

Corollary
Some languages are not Turing-recognizable.

- **Proof**:
  - Set of all strings on any alphabet is countable.
  - Set of all TMs is countable.
  - Set of all languages, **L**, is uncountable:
    - Set of all infinite binary sequences, **B**, is uncountable.
    - There is a correspondence between **L** and **B**:
      - Function f : **L** $\rightarrow \mathbf{B}$, where $\mathrm{f(A)}$ is the characteristic sequence of A is one-to-one and onto.
      - $\mathbf{B}$ is uncountable, thus $\mathbf{L}$ is uncountable.
  - We conclude that some languages are not recognized by any Turing machine.

Undecidability

Back to Theorem

- The acceptance problem for TMs:

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

Theorem
A_{TM} is undecidable.

- Proof: By contradiction. Suppose H is a decider for A_{TM}.

$$H(\langle M, w\rangle) = \begin{cases} \mathit{accept} & \text{if } M \text{ accepts } w \\ \mathit{reject} & \text{if } M \text{ does not accept } w. \end{cases}$$

- Construct another TM D:

D = "On input $\langle M\rangle$, where M is a TM:

1. Run H on input $\langle M, \langle M\rangle\rangle$.
2. Output the opposite of what H outputs. That is, if H accepts, *reject*; and if H rejects, *accept*."

- The acceptance problem for TMs:

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

Theorem
A_{TM} is undecidable.

- Proof: By contradiction. Suppose H is a decider for A_{TM}.

$$H(\langle M, w\rangle) = \begin{cases} \textit{accept} & \text{if } M \text{ accepts } w \\ \textit{reject} & \text{if } M \text{ does not accept } w. \end{cases}$$

- Construct another TM D:

$$D(\langle M\rangle) = \begin{cases} \textit{accept} & \text{if } M \text{ does not accept } \langle M\rangle \\ \textit{reject} & \text{if } M \text{ accepts } \langle M\rangle. \end{cases} \Longrightarrow D(\langle D\rangle) = \begin{cases} \textit{accept} & \text{if } D \text{ does not accept } \langle D\rangle \\ \textit{reject} & \text{if } D \text{ accepts } \langle D\rangle. \end{cases}$$

- Thus, neither TM D nor TM H can exist.

- The acceptance problem for TMs:

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

Theorem
A_{TM} is undecidable.

- Proof (summary):
  - H accepts $\langle M, w\rangle$ exactly when M accepts w.
  - D rejects $\langle M\rangle$ exactly when M accepts $\langle M\rangle$.
  - D rejects $\langle D\rangle$ exactly when D accepts $\langle D\rangle$.

Undecidability

Back to Theorem

- Did we use diagonalization in the proof?

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

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

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

A Turing-Unrecognizable Language

- Recall that the complement of a language is the language consisting of all strings that are not in the language.

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

A Turing-Unrecognizable Language

Theorem
A language is decidable iff it is Turing-recognizable and co-Turing-recognizable.

- Proof: Forward direction:
  - If language A is decidable, then it is also Turing-recognizable,
  - If language A is decidable, then $\bar{A}$ is also decidable, which is again Turing-recognizable.

A Turing-Unrecognizable Language

Theorem
A language is decidable iff it is Turing-recognizable and co-Turing-recognizable.

- Proof: Reverse direction:
  - Assume both languages A and $\bar{A}$ are Turing-recognizable,
  - Let M1 be the recognizer for A,
  - Let M2 be the recognizer for $\bar{A}$,
  - Construct a decider M for A:

$M =$ "On input w:

1. Run both M_1 and M_2 on input w in parallel.
2. If M_1 accepts, *accept*; if M_2 accepts, *reject*."

  - Every string w is either in A or $\bar{A}$.

A Turing-Unrecognizable Language

Corollary
$\overline{A_{TM}}$ is not Turing-recognizable.

- Proof:
    - Is A_{TM} Turing-recognizable.
    - If $\overline{A_{TM}}$ is Turing-recognizable, then A_{TM} should be decidable.
    - However, we proved that A_{TM} is not decidable.
    - Thus, $\overline{A_{TM}}$ must be Turing-unrecognizable.