



香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen



CSC 6011 - Theory of Computation - Lecture 9: Reduction

Time: Mon & Wed, 4:00 - 5:20 PM

Location: Teaching A, 107

Instructor: Tao LIN (林涛)

TA: Yefan GAO (高叶繁)

Fall 2026

Lecture 9 – Reduction

Last time:

- The Diagonalization method
- A_{TM} , $HALT_{TM}$ are undecidable
- $\overline{A_{TM}}$ is T-unrecognizable

Today:

- The Reduction method
 - for proving undecidability and T-unrecognizability.
- General reduction
- Mapping reduction

Reduction: An Example

If we know that some problem (say A_{TM}) is undecidable, we can use that fact to show other problems are undecidable.

Recall: $HALT_{TM} = \{\langle M, w \rangle \mid M \text{ halts on input } w\}$ is undecidable (proved by diagonalization)

Another proof: show that A_{TM} can be “reduced” to $HALT_{TM}$

(if $HALT_{TM}$ is decidable, then A_{TM} is decidable)

Let TM R decide $HALT_{TM}$.

Construct TM S to decide A_{TM} .

$S =$ “On input $\langle M, w \rangle$

1. Use R to test if M halts on w . If not, *reject*.
2. Now we know M halts on w . Simulate M on w .
3. If M accepts w , then *accept*.
If M rejects w , then *reject*.

TM S decides A_{TM} , a contradiction. Therefore $HALT_{TM}$ is undecidable.

Reduction: Formal Definition

Let A and B be two languages (or problems). We say A is reducible to B if we can use a solver for B to solve A . This process is called reduce A to B .

Example 1: Measuring the area of a rectangle is reducible to measuring the lengths of its sides.

Example 2: A_{NFA} is reducible to A_{DFA} .

Example 3: Homework 2 asks you to reduce A_{TM} to TWO_{TM} .
(whether a Turing machine accepts two strings)

If A is reducible to B , then solving B gives a solution to A .

- B is easy $\rightarrow A$ is easy.

- A is hard $\rightarrow B$ is hard.

this is the direction we will use

E_{TM} is undecidable

Theorem: $E_{TM} = \{ \langle M \rangle \mid M \text{ is a TM and } L(M) = \emptyset \}$ is undecidable

Prove by reducing A_{TM} to E_{TM} . (if E_{TM} is decidable, then A_{TM} is decidable)

Let TM R decide E_{TM} .

Construct TM S deciding A_{TM} .

$S =$ “On input $\langle M, w \rangle$

1. Transform M to a new TM:

1. If $x \neq w$, *reject*.

$M_w =$ “ On input x : 2. else run M on w

3. *Accept* if M accepts. ”

2. Use R to test whether $L(M_w) = \emptyset$

3. If YES (so M does not accept w), then *reject*.

If NO (so M accepts w), then *accept*.

M_w is a “filter”: always rejects input $x \neq w$. On input $x = w$, same as M

$$L(M_w) = \begin{cases} \{w\} & \text{if } M \text{ accepts } w \\ \emptyset & \text{if } M \text{ does not accept } w \end{cases}$$

Fun Quiz

Let Q = “ Will a PPT presentation ever display a "Bonus Question"? ”

Can we answer Q without going over the animations in the PPT file?

(a) Yes -- we can search if "Bonus Question" appears in the file.

(b) No -- Q is undecidable.

Proof:

PPT is Turing-Complete: PPT animations can simulate Turing machines [1].

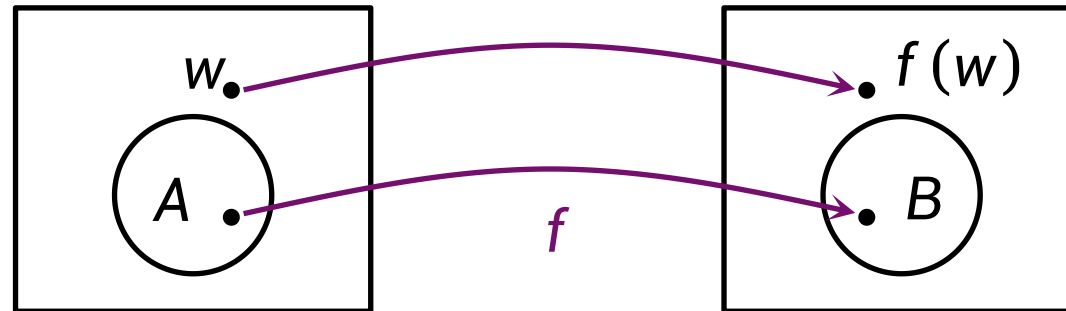
Hence, undecidable problems about Turing machines (such as A_{TM} and $HALT_{TM}$) can be reduced to PPT animation problems.

[1] Tom Wildenhain, “On the Turing Completeness of MS PowerPoint” (2017).

Mapping Reduction

Defn: Function $f: \Sigma^* \rightarrow \Sigma^*$ is computable if there is a TM F where F on input w halts with $f(w)$ on its tape, for all strings w .

Defn: A is mapping-reducible to B ($A \leq_m B$) if there is a computable function f (“reduction function”) where $w \in A$ iff $f(w) \in B$.



Example: $A_{\text{TM}} \leq_m \overline{E_{\text{TM}}}$

Recall: $M_w =$ “On input x :

1. If $x \neq w$, *reject*.
2. else run M on w
3. *Accept* if M accepts.”

The reduction function f is $f(\langle M, w \rangle) = \langle M_w \rangle$

Because $\langle M, w \rangle \in A_{\text{TM}}$ iff $\langle M_w \rangle \in \overline{E_{\text{TM}}}$
(M accepts w iff $L(M_w) \neq \emptyset$)

Mapping Reduction - properties

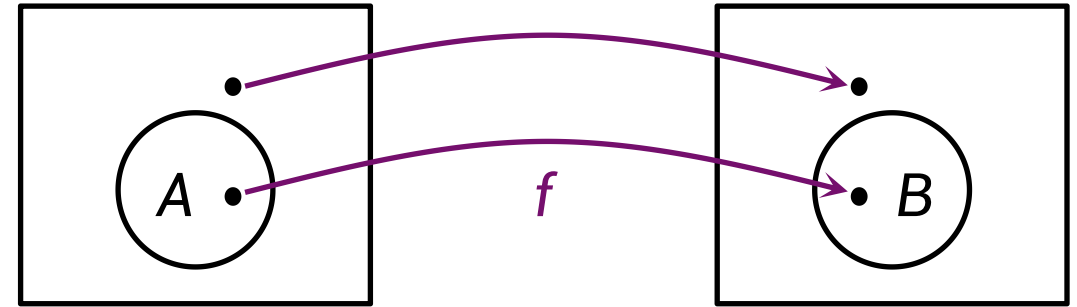
Theorem: If $A \leq_m B$ and B is decidable, then A is decidable.

Proof: Say TM R decides B .

Construct TM S deciding A :

$S =$ “On input w

1. Compute $f(w)$
2. Run R on $f(w)$ to test if $f(w) \in B$
3. If R halts then output same result.”



Corollary: If $A \leq_m B$ and A is undecidable, then B is undecidable.

Theorem: If $A \leq_m B$ and B is T-recognizable, then A is T-recognizable.

Proof: Same as above.

Corollary: If $A \leq_m B$ and A is T-unrecognizable, then B is T-unrecognizable.

Mapping Reduction - properties

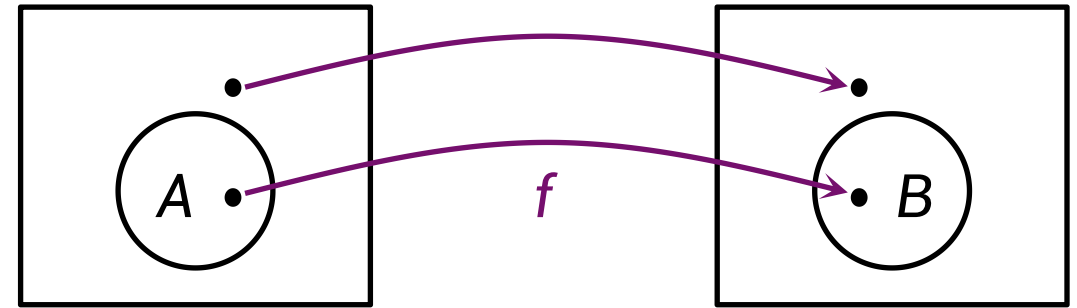
Theorem: If $A \leq_m B$ and B is decidable, then A is decidable.

Proof: Say TM R decides B .

Construct TM S deciding A :

$S =$ “On input w

1. Compute $f(w)$
2. Run R on $f(w)$ to test if $f(w) \in B$
3. If R halts then output same result.”



Corollary: If $A \leq_m B$ and A is undecidable, then B is undecidable.

Theorem: If $A \leq_m B$ and B is T-recognizable, then A is T-recognizable.

Proof: Same as above.

Corollary: If $A \leq_m B$ and A is T-unrecognizable, then B is T-unrecognizable.

Mini-Quiz 9.2

Suppose $A \leq_m B$, which of the following is true?

(a) $B \leq_m A$

(b) $\bar{A} \leq_m \bar{B}$

(c) None of the above

E_{TM} is T-unrecognizable

Recall $E_{\text{TM}} = \{ \langle M \rangle \mid M \text{ is a TM and } L(M) = \emptyset \}$

Theorem: E_{TM} is T-unrecognizable

Proof: We have shown $A_{\text{TM}} \leq_m \overline{E_{\text{TM}}}$ using the reduction function $f(\langle M, w \rangle) = \langle M_w \rangle$

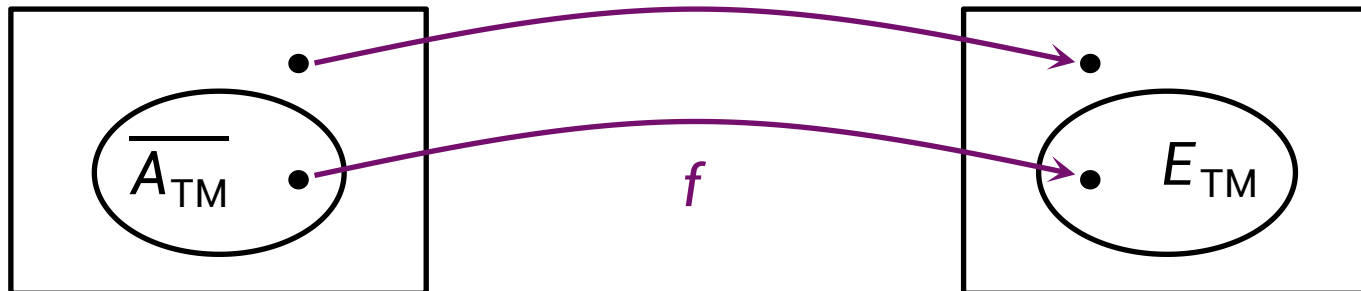
Taking complement on both sides, $\overline{A_{\text{TM}}} \leq_m E_{\text{TM}}$

Specifically: $\langle M, w \rangle \in \overline{A_{\text{TM}}}$ iff $\langle M_w \rangle \in E_{\text{TM}}$

M does not accept w iff $L(M_w) = \emptyset$

Recall: $M_w =$ “On input x

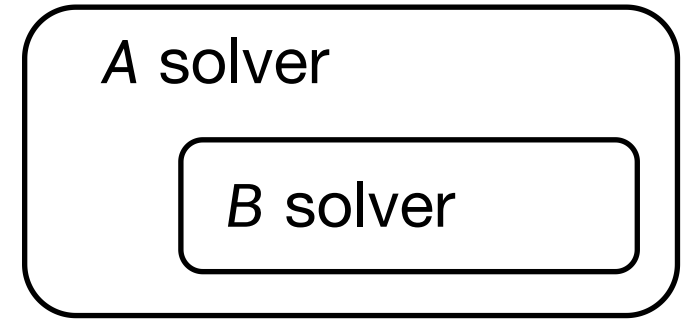
1. If $x \neq w$, *reject*.
2. else run M on w
3. *Accept* if M accepts.”



Mapping vs General Reductions

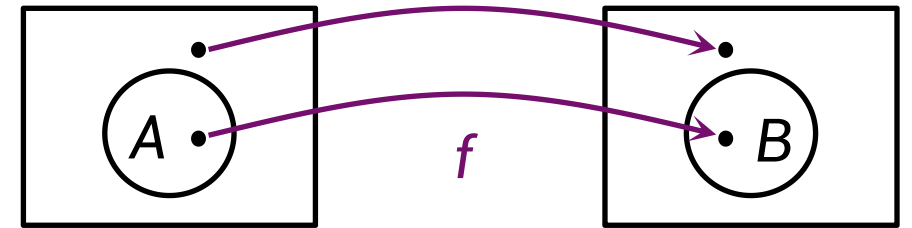
(General) Reduction of A to B : Use B solver to solve A .

- Make use the B solver freely (e.g., multiple times)
- Useful to prove undecidability



Mapping Reduction of A to B : Translate A -strings to B -strings.

- A special type of reduction
- Useful to prove T-unrecognizability



Noteworthy difference:

- A is reducible to $\bar{A}$ (Why?)
- A may be not mapping-reducible to $\bar{A}$.

For example $\overline{A_{TM}} \not\leq_m A_{TM}$ (Why?)

Reduction Proof - Templates

To prove B is undecidable:

- Show undecidable A is reducible to B . (often A is A_{TM})
- Template: Assume TM R decides B .
Construct TM S deciding A . Contradiction.

To prove B is T-unrecognizable:

- Show T-unrecognizable A is mapping-reducible to B . (often A is $\overline{A_{\text{TM}}}$)
- Template: give computable reduction function f .

EQ_{TM} and $\overline{EQ_{TM}}$ are T-unrecognizable

$$EQ_{TM} = \{ \langle M_1, M_2 \rangle \mid M_1 \text{ and } M_2 \text{ are TMs and } L(M_1) = L(M_2) \}$$

Theorem: Both EQ_{TM} and $\overline{EQ_{TM}}$ are T-unrecognizable

Proof: (1) $\overline{A_{TM}} \leq_m EQ_{TM}$

(2) $\overline{A_{TM}} \leq_m \overline{EQ_{TM}}$

For any w let $T_w =$ “On input x

1. Ignore x .

2. Simulate M on w .”

On all inputs, T_w acts in the same way as M on w .

(1) We give f which maps $\overline{A_{TM}}$ strings (of the form $\langle M, w \rangle$) to EQ_{TM} strings (of the form $\langle T_1, T_2 \rangle$).

$f(\langle M, w \rangle) = \langle T_w, T_{\text{reject}} \rangle$ T_{reject} is a TM that always rejects.

(2) Similarly $f'(\langle M, w \rangle) = \langle T_w, T_{\text{accept}} \rangle$ T_{accept} always accepts.

Direction of Reduction

When we reduce A to B , we show how to solve A by solving B .

This means that A is (weakly) less difficult than B . (Hence the $\leq_m$ notation)

Want to solve a problem: Reduce this problem to another (solvable) problem.

Prove that a problem is hard: Reduce another (difficult) problem to this problem.

Quick review of today

1. Introduced The Reduction Method to prove undecidability and unrecognizability.
2. Defined mapping reduction as a special type of reductions.
3. E_{TM} is undecidable (via general reduction).
4. E_{TM} is T-unrecognizable (via mapping reduction).
5. EQ_{TM} and $\overline{EQ_{TM}}$ are T-unrecognizable.

These slides are adapted from Michael Sipser, *18.404J Theory of Computation*, Fall 2020. Massachusetts Institute of Technology:

- Source: MIT OpenCourseWare <https://ocw.mit.edu>

Except for separately identified third-party material, this adapted presentation is licensed under [CC BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/).