بسم الله الرحمن الرحیم

Reducibility

Ali Shakiba
ali.shakiba@vru.ac.ir
Vali-e-Asr University of Rafsanjan

What we are going to discuss?

- Undecidable problems from language theory
    - Reductions via computation histories
- Mapping reducibility
    - Computable functions
    - Formal definition of mapping reducibility
- Post correspondence problem, or PCP

Reduction

A way of converting one problem to another problem in such a way that a solution to the second problem can be used to solve the first problem.

Reducing language A to language B means

- Using **deciders** of language B to **decide** language A.
- Using **recognizers** of language B to **recognize** language A.

Critical Point to Consider

Reducibility say nothing about solving A or B alone, but only about the solvability of A in the presence of a solution to B

Reduction of A to B

- Problem B is decidable
  - Then, problem A is also decidable.

Reduction of A to B

- Problem A is undecidable
  - Then, problem B is also undecidable.

So, to show that a problem is undecidable:

Reduce an undecidable problem to it

$$HALT_{TM} = \{\langle M, w\rangle | M \text{ is a TM and halts on input } w\}$$

Undecidable

Theorem 5.1

$HALT_{TM} = \{\langle M, w\rangle | M \text{ is a TM and halts on input } w\}$

- The proof is by contradiction.
  - Assume that $HALT_{TM}$ is decidable
  - Then, we reduce A_{TM} to $HALT_{TM}$
  - So, A_{TM} is decidable, too.
  - This is a contradiction.

$HALT_{TM} = \{\langle M, w\rangle | M \text{ is a TM and halts on input } w\}$

- Assume that there exists a TM R which decides $HALT_{TM}$
- We will construct a new TM S which decides A_{TM}
- What's halt?
  - Given an input $\langle M, w\rangle$, we need to print accept if M accepts w and print reject if M rejects w or loops.
- Reduce A_{TM} to $HALT_{TM}$
  - How?
- So, $HALT_{TM}$ is undecidable.

$HALT_{TM} = \{\langle M, w \rangle | M \text{ is a TM and halts on input } w\}$

- Assume that there exists a TM R which decides $HALT$
- We will construct a

$S =$ "On input $\langle M, w \rangle$, an encoding of a TM M and a string w:

1. Run TM R on input $\langle M, w \rangle$.
2. If R rejects, *reject*.
3. If R accepts, simulate M on w until it halts.
4. If M has accepted, *accept*; if M has rejected, *reject*."

undecidable.

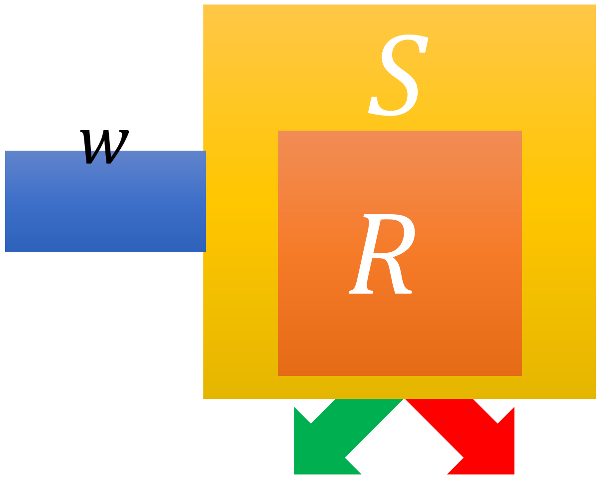
$$E_{TM} = \{\langle M\rangle | M \text{ is a TM and } L(M) = \emptyset\}$$

Undecidable

Theorem 5.2

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

- We will do that by reducing A_{TM} to E_{TM}.
  - Assume that R is a TM which decides E_{TM}.
  - Now, we construct a TM S which decides A_{TM} using R.



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

- S takes $\langle M, w \rangle$ as input,
- So, it creates a description of a new TM M_1 which is identical to M, except it checks if the input to M is w,
  - If the input is not w, then M_1 should reject.
  - If the input is w, then M_1 runs M on input w and accept if M does.
- Now, we have S as follows:

$S =$ "On input $\langle M, w \rangle$, an encoding of a TM M and a string w:

1. Use the description of M and w to construct the TM M_1 just described.
2. Run R on input $\langle M_1 \rangle$.
3. If R accepts, *reject*; if R rejects, *accept*."

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

Undecidable

Theorem 5.4

$$\text{Regular}_{TM} = \{\langle M \rangle | M \text{ is a TM and } L(M) \text{ is regular}\}$$

Undecidable

Theorem 5.3