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.

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

Undecidable

Theorem 5.1

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

Undecidable

Theorem 5.2

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

Undecidable

Theorem 5.3

Exercise 5.28
(Rice's theorem)

$S =$ "On input $\langle M, w \rangle$, where M is a TM and w is a string:

1. Construct the following TM M_2.

   $M_2 =$ "On input x:

   1. If x has the form $0^n 1^n$, *accept*.
   2. If x does not have this form, run M on input w and *accept* if M accepts w."

2. Run R on input $\langle M_2 \rangle$.
3. If R accepts, *accept*; if R rejects, *reject*."

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

Undecidable

Theorem 5.4

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

1. Run R on input $\langle M, M_1 \rangle$, where M_1 is a TM that rejects all inputs.
2. If R accepts, *accept*; if R rejects, *reject*."

Computation Histories for Turing Machines

The sequence of configurations that the machine goes through as it processes the input.

Sequence of configurations $C_1, \dots, C_\ell$ is an **accepting computation history** for M on w where C_1 is the **start configuration** and C_ℓ is an **accepting configuration** and every configuration C_{i+1} legally follows from configuration C_i.

Computation Histories for Turing Machines

The sequence of configurations that the machine goes through as it processes the input.

Sequence of configurations $C_1, \ldots, C_\ell$ is a **rejecting computation history** for M on w where C_1 is the **start configuration** and C_ℓ is an **rejecting configuration** and every configuration C_{i+1} legally follows from configuration C_i.

Computation Histories for Turing Machines

$\ell < \infty$

Sequence … M on w
where C_1 i… every
configurat…

If TM M does not reject on input w, then …

No accepting or rejecting computation history exists

Computation histories for …

- Deterministic machines
  - at most one computation history on any given input
- Non-deterministic machines
  - many computation histories are possible

$$A_{LBA} = \{\langle M, w\rangle | M \text{ is an LBA that accepts string } w\}$$

What's LBA?

Linear Bounded Automaton, or LBA, is …

- a restriction of a TM in terms of memory,
- the tape head is not permitted to move off the portion of the tape containing the input.
- the tape head stays on the rightmost or leftmost tape cell if the machine tries to move off the end of the input.

Why the name "Linear"?

$A_{LBA} = \{\langle M, w\rangle | M \text{ is an LBA that accepts string } w\}$

Decidable

Theorem 5.9

Lemma 5.8: There are qng^n distinct configurations for an LBA with q states and g symbols on input of length n.

Proof on board

$A_{LBA} = \{\langle M, w\rangle | M \text{ is an LBA that accepts string } w\}$

Decidable

Theorem 5.9

$L =$ "On input $\langle M, w\rangle$, where M is an LBA and w is a string:

1. Simulate M on w for qng^n steps or until it halts.
2. If M has halted, *accept* if it has accepted and *reject* if it has rejected. If it has not halted, *reject*."

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

Undecidable

Theorem 5.10

$S =$ "On input $\langle M, w\rangle$, where M is a TM and w is a string:

1. Construct LBA B from M and w as described in the proof idea.
2. Run R on input $\langle B\rangle$.
3. If R rejects, *accept*; if R accepts, *reject*."

$$ALL_{CFG} = \{\langle G \rangle | G \text{ is a CFG and } L(G) = \Sigma^*\}$$

Undecidable

Theorem 5.13

Do not forget EQ_{CFG} is undecidable (Exercise 5.1).