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

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

Undecidable

Theorem 5.13

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

Post Correspondence Problem, or PCP

$\neq$ **P**robabilistically **C**heckable **P**roof

$$\left\{ \left[\frac{\mathtt{b}}{\mathtt{ca}}\right], \left[\frac{\mathtt{a}}{\mathtt{ab}}\right], \left[\frac{\mathtt{ca}}{\mathtt{a}}\right], \left[\frac{\mathtt{abc}}{\mathtt{c}}\right] \right\}$$

matching

$$\left[\frac{\mathtt{a}}{\mathtt{ab}}\right]\left[\frac{\mathtt{b}}{\mathtt{ca}}\right]\left[\frac{\mathtt{ca}}{\mathtt{a}}\right]\left[\frac{\mathtt{a}}{\mathtt{ab}}\right]\left[\frac{\mathtt{abc}}{\mathtt{c}}\right]$$

$$\begin{array}{c} \mathtt{a\ b\ c\ a\ a\ a\ b\ c} \\ \mathtt{a\ b\ c\ a\ a\ a\ b\ c} \end{array}$$

$PCP = \{\langle P \rangle | P \text{ is an instance of PCP with a match}\}$

Undecidable

Theorem 5.15

$PCP = \{\langle P \rangle | P \text{ is an instance of PCP with a match}\}$

- We reduce it from A_{TM} with computation histories
  - Given a TM M and an input w, we construct an instance of PCP P where M accepts w if there is a match in P.
- How can we construct P so that a match is an accepting computation history for M on w?
  - Each domino links a position or positions in one configuration with the corresponding ones in the next configuration.
  - Some simplifications:

MPCP

1. TM M on input w never tries to move its head off the left-hand end of tape
2. If $w = \varepsilon$, the string $\sqcup$ is used in place of w in the construction
3. PCP is modified such that a match must start with the first domino $\left[\frac{t_1}{b_1}\right]$.

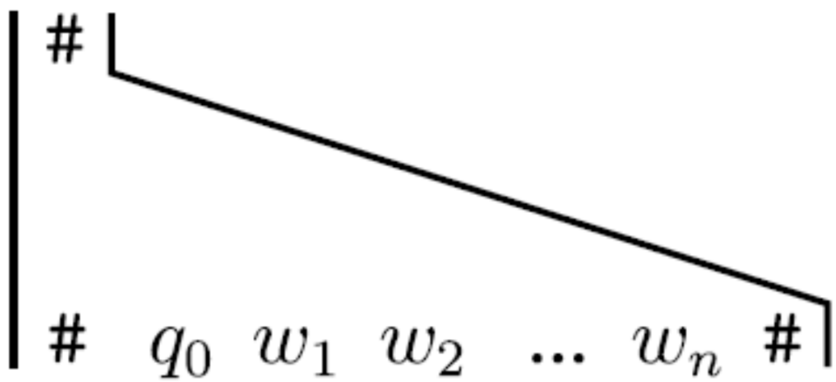
$PCP = \{\langle P \rangle | P \text{ is an instance of PCP with a match}\}$

- Assume TM R decides PCP, then we construct a TM S which decides A_{TM}.
  - Let $M = \left(Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}}\right)$
  - TM S needs to construct an instance of PCP P that has a match iff M accepts w.
    - At first, S constructs and instance of MPCP P'

MPCP P'

1. The first domino is

$$\left[\frac{t_1}{b_1}\right] = \left[\frac{\#}{\#q_0 w_1 w_2 \ldots w_n \#}\right]$$



MPCP P'

2. For every $a, b \in \Gamma$ and every $q, r \in Q$ where $q \neq q_{\text{reject}}$, if $\delta(q, a) = (r, b, \boldsymbol{R})$, put $\left[\frac{qa}{br}\right]$ into P'.
3. For every $a, b \in \Gamma$ and every $q, r \in Q$ where $q \neq q_{\text{reject}}$, if $\delta(q, a) = (r, b, \boldsymbol{L})$, put $\left[\frac{cqa}{rcb}\right]$ into P'.
4. For every $a \in \Gamma$, put $\left[\frac{a}{a}\right]$ into P'.
5. Put $\left[\frac{\#}{\#}\right]$ and $\left[\frac{\#}{\sqcup\#}\right]$ into P'.

MPCP P'

6. For every $a \in \Gamma$, put $\left[\frac{a q_{\text{accept}}}{q_{\text{accept}}}\right]$ and $\left[\frac{q_{\text{accept}} a}{q_{\text{accept}}}\right]$ into P'.
7. Finally, add the following domino into P':

$$\frac{q_{\text{accept}}\#\#}{\#}$$

MPCP P'

- The MPCP instance P' constructed so far has a match iff M accepts w.
- However, if we treat it as a PCP instance, it always has a match …
  - What's that match?
- We need to convert MPCP P' into an instance of PCP P where a match in P exists iff M accepts w.
  - How?

Our trick

- For string $u = u_1 u_2 \dots u_n$, we define:
  - $\star u = * u_1 * u_2 * \cdots * u_n$
  - $\star u \star = * u_1 * u_2 * \cdots * u_n *$
  - $u \star = u_1 * u_2 * \cdots * u_n *$
- For MPCP instance $P' = \left\{ \frac{t_1}{b_1}, \frac{t_2}{b_2}, \frac{t_3}{b_3}, \dots, \frac{t_k}{b_k} \right\}$, we construct the following PCP instance

$$P = \left\{ \frac{\star t_1}{\star b_1 \star}, \frac{\star t_1}{b_1 \star}, \frac{\star t_2}{b_2 \star}, \frac{\star t_3}{b_3 \star}, \dots, \frac{\star t_k}{b_k \star}, \frac{* \bowtie}{\bowtie} \right\}.$$

$PCP = \{\langle P \rangle | P \text{ is an instance of PCP with a match}\}$

Undecidable

Theorem 5.15

Mapping reducibility, a.k.a. many-to-one reducibility

- Reducing problem A to B by mapping reducibility means:
  - There exists a **computable function** which converts instances of problem A to instances of problem B.
- A function $f: \Sigma^* \to \Sigma^*$ is a **computable function** if some Turing machine M on every input w halts with just $f(w)$ on its tape.
- Language A is mapping reducible to language B, written $A \leq_m B$, if there is a computable function $f: \Sigma^* \to \Sigma^*$ where for every $w \in \Sigma^*$
$$w \in A \Leftrightarrow f(w) \in B$$
  - The function f is called the reduction from A to B.

Some Results

Theorem 5.22: If $A \leq_m B$ and B is decidable, then A is decidable.

Corollary 5.23: If $A \leq_m B$ and A is undecidable, then B is undecidable.

Theorem 5.28: If $A \leq_m B$ and B is Turing-recognizable, then A is Turing recognizable.

Corollary 5.29: If $A \leq_m B$ and A is not Turing-recognizable, then B is not Turing-recognizable.

Theorem 5.30

EQ_{TM} is neither Turing-recognizable nor co-Turing-recognizable.

- EQ_{TM} is not Turing-recognizable:
  - Reducing from A_{TM} to $\overline{EQ_{TM}}$.

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

1. Construct the following two machines, M_1 and M_2.

$M_1 =$ "On any input:

1. *Reject*."

$M_2 =$ "On any input:

1. Run M on w. If it accepts, *accept*."

2. Output $\langle M_1, M_2\rangle$."

EQ_{TM} is neither Turing-recognizable nor co-Turing-recognizable.

- $\overline{EQ_{TM}}$ is not Turing-recognizable:
  - Reducing from A_{TM} to EQ_{TM}.

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

1. Construct the following two machines, M_1 and M_2.

$M_1 =$ "On any input:

1. *Accept*."

$M_2 =$ "On any input:

1. Run M on w.

2. If it accepts, *accept*."

2. Output $\langle M_1, M_2 \rangle$."

Do not forget the MIDTERM on Aban 10th , 1395

- Chapters 3 to 5 of Sipser's TOC book.
- Two parts:
  1. Closed book: in-class
  2. Take home exam: you have 24 hours to return the answers.