Lecture 3

Reducibility :

- $HALT_{TM} = \{\langle M,w\rangle : M$ is a TM and M halts on input $w\}$ is undecidable.

Proof:- Assume for contradiction that $HALT_{TM}$ is decidable. Let R be a decider for $HALT_{TM}$. Let

S = " On input $\langle M,w\rangle$:

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."

S decides A_{TM}. Contradiction □.

- $E_{TM} = \{\langle M\rangle : M$ is a TM and $L(M) = \phi\}$. is undecidable.

Proof: For TM M and string w, let:

$M_{(M,w)}$ = " On input x:

1) If $x \neq w$ reject.
2) Otherwise run M on input w.
3) Accept if M accepts."

Note that $L(M_{(M,w)}) \neq \phi \iff$ M accepts w.

Assume for contradiction that E_{TM} is decidable. Let R be a decider for E_{TM}. Let

S = " On input $\langle M,w\rangle$:

1. Construct description $\langle M_{(M,w)}\rangle$ of TM machine M_w.
2. Run R on input $\langle M_{(M,w)}\rangle$
3. If R accepts, reject; if R rejects, accept."

S decides A_{TM}. Contradiction □.

- $REGULAR_{TM} = \{\langle M\rangle : M$ is a TM and $L(M)$ is a regular language$\}$. is undecidable.

Proof:- Assume for contradiction that R decides $REGULAR_{TM}$. Let

$S =$ " On input $\langle M, w\rangle$:

1. Construct the following TM M_1.

   $M_w =$ " On input x:

   1. If x has the form $0^n 1^n$, accept.

   2. Otherwise run M on input w. Accept if M accepts w."

2. Run R on input $\langle M_w \rangle$.

3. If R accepts, accept; if R rejects, reject."

S decides A_{TM}. Contradiction □.

Def:- Property of languages: Property P is the language consisting of TM descriptions such that

$\forall M_1, M_2 : \left(L(M_1) = L(M_2) \right) \implies \left(\langle M_1 \rangle \in P \iff \langle M_2 \rangle \in P \right)$.

Property P is non-trivial if it contains some and not all TM descriptions.

Rice's Theorem:- Every non-trivial property of languages is undecidable.

Proof:- Let P be a non-trivial property. Assume for contradiction that R_P decides P.

Let T_ϕ be a TM that always rejects, hence $L(T_\phi) = \phi$.

Assume without loss of generality (w.l.o.g.) that $\langle T_\phi \rangle \notin P$ (otherwise work with $\overline{P}$).

Since P is non-trivial assume $\langle T \rangle \in P$.

Let $S =$ " On input $\langle M, w \rangle$:

1. Construct description $\langle M_w \rangle$ of the following TM M_w:

   $M_w =$ " On input x:

   1. Simulate M on w. If it halts and rejects, reject.
      If it accepts, go to stage 2.

   2. Simulate T on x. If T accepts x, accept."

2. Run R_P on input $\langle M_w \rangle$. If R_P accepts, accept. If R_P rejects, reject."

Note that M_w simulates T if M accepts w. Hence $L(M_w) = L(T)$ if M accepts w and $L(M_w) = \phi$ otherwise. Therefore $\langle M_w \rangle \in P \iff$ (M accepts w).

Hence S decides A_{TM}. Contradiction □.

- $EQ_{TM} = \{ \langle M_1, M_2 \rangle :$ M_1 and M_2 are TMs and $L(M_1) = L(M_2) \}$ is undecidable.

Proof:- Assume for contradiction that R decides EQ_{TM}.

Let $S =$ " On input $\langle M \rangle$:

1. Let M_1 be a TM such that $L(M_1) = \phi$

1. Let M_1 be a TM such that $L(M_1) = \phi$.
Run R on input $\langle M, M_1 \rangle$.

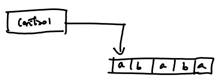
2. If R accepts, accept; if R rejects, reject."

S decides E_{TM}. Contradiction. $\square$

Reductions via Computation Histories:

Defn:- Let M be a Turing machine and w an input string. An accepting configuration history for M on w is a sequence of configurations $C_1, C_2, \ldots, C_k$, where C_1 is the start configuration of M on input w, C_k is an accepting configuration of M, and each C_i legally follows from C_{i-1} according to the rules of M. A rejecting computation history for M on w is defined similarly except that C_k is a rejecting configuration.

Defn: A linear bounded automaton is a Turing machine that does not use more space than the length of the input



Examples 1) Deciders for A_{DFA}, A_{CFG}, E_{DFA}, E_{CFG}.

2) Every CFA can be decided by an LBA.

- $A_{LBA} = \{\langle M, w \rangle : M$ is an LBA and M accepts $w\}$ is decidable.

Lemma: Let M be an LBA with q states and g symbols in the tape alphabet. There are exactly $q n g^n$ distinct configurations of M for a tape of length n.

Proof: Easy to verify. $\square$

Thm: A_{LBA} is decidable.

Proof:- Decider for A_{LBA}

S = " On input $\langle M, w \rangle$:

1. Simulate M on w for $q n g^n$ steps or until it halts. ($n = |w|$).

2. If M accepts, accept. If M rejects, reject. If M has not halted reject."

- $E_{LBA} = \{\langle M \rangle : \langle M \rangle$ is LBA and $L(M) = \phi\}$ is undecidable.

Proof:- For TM M and string w consider language

$$L_{(M,w)} = \{\#C_1\#C_2\#\cdots\#C_k\# : C_1, C_2, \ldots C_k \text{ is an accepting computation history of } M \text{ on input } w.\}$$

Observe that for any (M,w), $L_{(M,w)}$ is decidable language and can be decided by an LBA $B_{(M,w)}$.

Note also that $(L_{(M,w)} \neq \phi) \iff (M \text{ accepts } w) \iff ((M,w) \in A_{TM})$

Assume for contradiction that E_{LBA} is decidable. Construct TM S deciding A_{TM} as follows:

Assume for contradiction that E_{LBA} is decidable. Construct TM S deciding A_{TM} as follows:

S = " On input $\langle M, w \rangle$:

1. Construct LBA $B_{(M,w)}$.

2. Run T (decider for E_{LBA}) on input $B_{(M,w)}$.

3. If T accepts, reject ; if T rejects, accept ."

S decides A_{TM} . Contradiction . $\square$.

- $ALL_{CFG} = \{ \langle G \rangle : G$ is a CFG and $L(G) = \Sigma^* . \}$ is undecidable.

Proof: For TM M and string w, let

$$L_{(M,w)} = \{ \# c_1 \# c_2^R \# c_3 \# c_4^R \# \dots \# c_k : c_1, c_2, \dots, c_k \text{ is NOT an accepting configuration of } M \text{ on input } w . \}$$

Note that $(L_{(M,w)} = \Sigma^*) \iff$ (M does NOT accept w) $\iff$ ($(M, w) \notin A_{TM}$)

Note also that for any (M, w), $L_{(M,w)}$ is decidable language and infact can be decided by a PDA $B_{(M,w)}$.

Assume for contradiction that ALL_{CFG} is decidable . Let T be a decider for ALL_{CFG}.

Construct S deciding A_{TM} as follows :

S = " On input (M, w) :

1. Construct PDA $B_{(M,w)}$ deciding $L_{(M,w)}$.

2. Construct grammer G of $B_{(M,w)}$.

3. Run T on input $\langle G \rangle$.

4. If T accepts, reject ; if T rejects, accept . "

S decides A_{TM} . Contradiction $\square$

Mapping Reducibility :-

Defn (Computable function) : 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.

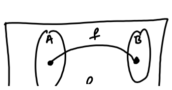
e.g. Add : $\langle m, n \rangle \to \langle m + n \rangle$
Multiply : $\langle m, n \rangle \to \langle m \times n \rangle$
Devide : $\langle m, n \rangle \to \langle \frac{m}{n} \rangle$ etc.

Defn : (Mapping reducible) : Language A is mapping reducible to language B, written $A \leq_m B$, if there is a computable function $f : \Sigma^* \to \Sigma^*$, such that for every w :

$w \in A \iff f(w) \in B$

The function is called the reduction of A to B.

Mapping reduction is also called as ' Many - to - one reduction '.



Thm: If $A \le_m B$ then (B is decidable) $\Rightarrow$ (A is decidable)

Proof:- Let M be a decider for B and let f be a reduction from A to B.

Decider N for A:

N = " On input w:

1. Compute $f(w)$.

2. Run M on $f(w)$ and output whatever M outputs." □

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

Example: a) $A_{TM} \le_m HALT_{TM}$.

F = " On input $\langle M, w \rangle$:
0. If input is not in correct form output the input.
1. Construct the following machine M'.

M' = " On input x:

1. Run M on x.
2. If M accepts, accept.
3. If M rejects, enter a loop."

2. Output $\langle M', w \rangle$."

b) $E_{TM} \le EQ_{TM}$. f: $\langle M \rangle \rightarrow \langle M, M_1 \rangle$ where M_1 rejects all inputs.

c) $A_{TM} \le \overline{E_{TM}}$ f: $\langle M, w \rangle \rightarrow \langle M_{(M,w)} \rangle$ (as in the proof earlier)

Thm:- If $A \le_m B$ and B is Turing-recognizable then A is Turing-recognizable.

Corollary: If $A \le_m B$ and A is not Turing-recognizable then B is not Turing-recognizable.

Thm: $A \le_m B$ implies $\overline{A} \le_m \overline{B}$

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

Proof: $A_{TM} \le \overline{EQ_{TM}}$.

F = " On input $\langle M, w \rangle$:

1. Construct following TMs M_1 and M_2.

M_1 = " On input x:
1. Reject."

M_2 = " On input x:
1. Run M on w. If it accepts, accept."

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

$A_{TM} \le EQ_{TM}$

G = " On input $\langle M, w \rangle$:

1. Construct TMs M_1 and M_2.

M_1 = " On input x:
1. Accept ."

M_2: " On input x:
1. Run M on w.
2. If M accepts, accept."

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

Post Correspondence Problem:

An instance of PCP is a collection of dominos.

$$P = \{ [\tfrac{t_1}{b_1}], [\tfrac{t_2}{b_2}], [\tfrac{t_3}{b_3}], \dots\dots, [\tfrac{t_k}{b_k}] \}$$

A match is a sequence $i_1, i_2, \dots, i_r$ such that $t_{i_1} t_{i_2} \dots t_{i_r} = b_{i_1} b_{i_2} \dots b_{i_r}$.

E.g. $P = \{ [\tfrac{b}{ca}], [\tfrac{a}{ab}], [\tfrac{ca}{a}], [\tfrac{abc}{c}] \}$

A match $[\tfrac{a}{ab}] [\tfrac{b}{ca}] [\tfrac{ca}{a}] [\tfrac{a}{ab}] [\tfrac{abc}{c}]$

Top string = Bottom string = abcaaabc .

$PCP = \{ \langle P \rangle$: P is a collection of dominos with a match $\}$.

Thm: PCP is undecidable.

Proof: We show reduction from A_{TM} via accepting computation histories

For any TM M and string w, we construct instance P of PCP where a match is an accepting computation history for M on w.

Simplifying assumptions: 1) M on w never attempts to move its head off the left hand end of the tape.

2) If $w = \varepsilon$, we use $w = \sqcup$ in the construction of P.

3) We require that match starts with first domino $[\tfrac{t_1}{b_1}]$ in P.

$MPCP = \{ \langle P \rangle$: P is a collection of dominos with a match that start with the first domino. $\}$

We first show $A_{TM} \leq_m MPCP$.

Let $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$. Let $w = w_1 w_2 \dots w_n$.

Part 1: Put $[\tfrac{\#}{\# q_0 w_1 w_2 \dots w_n \#}]$ as first domino in P.

Part 2: For $a, b \in \Gamma$ and every $q, r \in Q$ where $q \neq q_{reject}$,

if $\delta(q, a) = (r, b, R)$, put $[\tfrac{qa}{br}]$ into P.

Part 3: For every $a, b, c \in \Gamma$ and every $q, r \in Q$, $q \neq q_{reject}$,

if $\delta(q, a) = (r, b, L)$, put $[\tfrac{cqa}{rcb}]$ into P.

Part 4: For every $a \in \Gamma$, put $[\tfrac{a}{a}]$ into P.

Part 5: Put $[\tfrac{\#}{\#}]$ and $[\tfrac{\#}{\sqcup \#}]$ into P.

Part 5: Put $\left[\frac{\#}{\#}\right]$ and $\left[\frac{\#}{\sqcup\#}\right]$ into P.

Part 6: For every $a \in \Gamma$, put

$$\left[\frac{a\, q_{accept}}{q_{accept}}\right] \text{ and } \left[\frac{q_{accept}\, a}{q_{accept}}\right] \text{ into } P.$$

Part 7: Add $\left[\frac{q_{accept}\ \#\#}{\#}\right]$ into P.

Match beginning, let $w = 0100$.

$$\begin{array}{l} \# \\ \#\, q_0\, 0\, 1\, 0\, 0\, \# \end{array}$$

Let $\delta(q_0, 0) = (q_7, 2, R)$, so we have domino $\left[\frac{q_0\, 0}{2\, q_7}\right]$ in P.

We also have $\left[\frac{0}{0}\right]$, $\left[\frac{1}{1}\right]$, $\left[\frac{2}{2}\right]$ and $\left[\frac{\sqcup}{\sqcup}\right]$ in P.

$$\begin{array}{l} \#\ q_0\, 0\ \ 1\ \ 0\ \ 0\ \ \# \\ \#\, q_0\, 0\, 1\, 0\, 0\, \#\ \ 2\ q_7\ \ 1\ \ 0\ \ 0\ \ \# \end{array}$$

Let $\delta(q_7, 1) = (q_9, 2, L)$; we have $\left[\frac{2\, q_7\, 1}{q_9\, 2\, 2}\right]$ in P.

$$\cdots\cdots \quad \begin{array}{l} \#\ 2\ q_7\ 1\ \ 0\ \ 0 \\ \#\, 2\, q_7\, 1\, 0\, 0\ \ q_9\ 2\ 2\ \ 0\ \ 0 \end{array}$$

Suppose we reach q_{accept}

$$\cdots\cdot \quad \begin{array}{l} \#\ 2\ 1\ q_{accept}\ 0\ 2\ \# \\ \#\, 2\, 1\, q_{accept}\, 0\, 2\, \#\ \ 2\ 1\ q_{accept}\ 2\ \# \end{array} \quad \cdots\cdots \quad \begin{array}{l} \# \\ \#\ q_{accept}\ \# \end{array}$$

$$\cdots\cdot \quad \begin{array}{l} \#\ \ q_{accept}\ \#\ \# \\ \#\, q_{accept}\, \#\ \ \# \end{array}$$

Converting instance P of MPCP to instance P' of PCP

For string $u = u_1 \cdots u_n$, let

$$\star u = \star u_1 \star u_2 \cdots \star u_n \ ; \quad u\star = u_1 \star u_2 \star \cdots u_n \star \ ; \quad \star u \star = \star u_1 \star u_2 \cdots \star u_n \star.$$

Let $P = \left\{ \left[\frac{t_1}{b_1}\right], \left[\frac{t_2}{b_2}\right], \ldots, \left[\frac{t_k}{b_k}\right] \right\}$ instance starting with $\left[\frac{t_1}{b_1}\right]$.

Then let $P' = \left\{ \left[\frac{\star t_1}{\star b_1 \star}\right], \left[\frac{\star t_1}{b_1 \star}\right], \left[\frac{\star t_2}{b_2 \star}\right], \ldots, \left[\frac{\star t_k}{b_k \star}\right], \left[\frac{\star \Diamond}{\Diamond}\right] \right\}$.