Lecture 3

Reducibility:

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

  Proof: Assume for contradiction that $\text{HALT}_m$ is decidable. Let $R$ be a decider for $\text{HALT}_m$. Let $S$ be a TM that decides $\text{HALT}_m$.
  
  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 $\text{HALT}_m$. Contradiction. $\square$

- $\text{Em}_m = \{ \langle M, w \rangle : M \text{ is a TM and } L(M) = \emptyset \}$ is undecidable.

  Proof: For TM $M$ and string $w$, let $M_{\text{new}}$ be a TM that:

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

  Note that $L(M_{\text{new}}) \neq \emptyset \Leftrightarrow M$ accepts $w$.

  Assume for contradiction that $\text{Em}_m$ is decidable. Let $R$ be a decider for $\text{Em}_m$. Let $S_o$ be a TM that decides $\text{Em}_m$.

  1. Construct description $\langle M_{\text{new}} \rangle$ of TM machine $M_{\text{new}}$.
  2. Run $R$ on input $\langle M_{\text{new}} \rangle$.
  3. If $R$ accepts, reject; if $R$ rejects, accept.

  $S_o$ decides $\text{Em}_m$. Contradiction. $\square$

- $\text{REGULAR}_m = \{ \langle M, L \rangle : M \text{ is a TM and } L(M) \text{ is a regular language} \}$ is undecidable.
Proof. Assume for contradiction that \( R \) decides \( \text{REWRITE}_m \). Let

\[ S = \text{"On input } \langle M, w \rangle \text{:} \]

1. Run \( M \) on input \( w \).

\[ M_0 = \text{"On input } x \text{:} \]

3. If \( x \) has the form \( 0^n1^n \), accept.

2. Otherwise, run \( M \) on input \( w \). Accept if \( M \) accept \( w \).

1. Run \( R \) on input \( \langle M_0 \rangle \).

3. If \( R \) accept, accept; if \( R \) reject, reject.

\( \) decides \( \text{REWRITE}_m \). Contradiction.

Definition 2: Property \( P \) is a non-trivial property of languages if it is neither true nor false for all TMs.

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

Proof. Let \( P \) be a non-trivial property. Assume for contradiction that \( R \) decides \( P \).

Let \( T_q \) be a TM that always rejects, i.e., \( L(T_q) = \emptyset \).

Assume without loss of generality \( \langle T_q \rangle \notin P \). Otherwise, \( \langle T_q \rangle \in P \).

Since \( P \) is non-trivial, assume \( \langle T_q \rangle \notin P \).

Let \( S = \text{"On input } \langle M, w \rangle \text{:} \)

1. Construct description \( \langle M_0 \rangle \) of the following TM \( M_0 \):

\[ M_0 = \text{"On input } x \text{:} \]

1. Simulate \( M \) on \( w \). If it halts and rejects, reject.

2. If it accepts, go to step 2.

2. Simulate \( T_q \) on \( x \). If \( T_q \) accept \( x \), accept.

1. Run \( R \) on input \( \langle M_0 \rangle \). If \( R \) accept, accept. If \( R \) reject, reject.

Note that \( M_0 \) simulates \( T_q \) if \( M \) accept \( w \). Hence \( L(M_0) = L(T_q) \) if \( M \) accept \( w \) and \( L(w) = \emptyset \) otherwise.

Therefore \( \langle M_0 \rangle \notin P \). Hence \( S \) decides \( \text{REWRITE}_m \). Contradiction.

\( \) \( \text{EQ}_m = \{ \langle M, M' \rangle : M \text{ and } M' \text{ are TMs and } L(M) = L(M') \} \) is undecidable.

Proof. Assume for contradiction that \( R \) decides \( \text{EQ}_m \).

Let \( S = \text{"On input } \langle M, M' \rangle \text{:} \)

1. If \( M, M' \text{ are TMs such that } L(M) \neq L(M') \)
1. Let \( M \) be a TM such that \( L(M) \neq \phi \).
   
   Run \( R \) on input \( L(M) \).

2. If \( R \) accepts, accept; if \( R \) rejects, reject.

3. Decide \( \exists w \in L(M) \).

Reduction via Computation Histories:

Define: Let \( M \) be a Turing machine and \( w \) an input string. An accepting configuration history for \( M \) on \( w \) is a sequence of configurations \( C_0, C_1, \ldots, C_k \), where \( C_0 \) 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.

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

Example 3) Deciders for \( \exists \text{acc} \); \( \exists w \in L(M) \).

2) Every LBA can be decided by an LBA.

\( \exists \text{acc} = \{ \langle M, w \rangle : M \text{ is an LBA and } M \text{ accepts } w \} \) is decidable.

Lemma: Let \( M \) be a LBA with \( q \) states and \( q \) symbols in the tape alphabet. Then there are exactly \( q^q \) distinct configurations of \( M \) for a tape of length \( n \).

Proof: Easy to verify. \( \square \)

Then: \( \exists \text{acc} \) is decidable.

Proof: Decider for \( \exists \text{acc} \)

1. \( \langle M, w \rangle \) as input:

2. Simulate \( M \) on \( w \) for \( q^q \) steps or until it halts. ( \( < w \), \( < w \) ).

3. If \( M \) accepts, accept. If \( M \) rejects, reject. If \( M \) has not halted reject.

\( \exists \text{enc} = \{ \langle M \rangle : \langle M \rangle \text{ is an LBA and } L(M) \neq \phi \} \) is undecidable.

Proof: For \( \exists \text{acc} \) and string \( w \) consider language:

\( L(w, \alpha) = \{ \langle M \rangle \text{ s.t. } C_\alpha \text{ is an accepting configuration history of } M \text{ on input } w \} \).

Observe that for any \( \langle M \rangle \), \( \exists \text{acc} \langle M \rangle \) is decidable language and can be decided by an LBA. Hence:

Note that \( (L(w, \alpha) \neq \emptyset) \iff (M \text{ accepts } \langle M \rangle \iff (M) \in \text{ acc}) \)

Assume for contradiction that \( \exists \text{enc} \) is decidable. Construct TM \( S \) deciding \( \exists \text{enc} \) as follows:
Assume for contradiction that $E_{\text{dec}}$ is decidable. Construct TM $S$ deciding $A_{\text{dec}}$ as follows:

1. On input $<w>$:
   1. Construct PDA $B(w)$.
   2. Run $T$ (decider for $E_{\text{dec}}$) on input $B(w)$.
   3. If $T$ accepts, reject; if $T$ rejects, accept.

$S$ decides $A_{\text{dec}}$. Contradiction.

- $A_{\text{dec}} = \{<b> : \text{ } b \in \{0,1\}^* \text{ and } L(b) = \emptyset \text{ is undecidable.}\}

Proof: For TM $T$ and string $w$, let

$L(w) = \{<b> : \text{ } b \in \{0,1\}^* \text{ and } \text{ } L(b) = \emptyset \text{ in } \text{Nat} \text{ an accepting configuration of } M \text{ on input } w\}$

Note that $(L(w) = \emptyset) \iff (M \text{ does not accept } w) \iff (\langle w \rangle \notin A_{\text{dec}})$

Note also that for any $(\langle w \rangle, L(w))$, $L(w)$ is decidable, hence and input can be decided by a PDA $B(w)$.

Assume for contradiction that $A_{\text{dec}}$ is decidable. Let $T$ be a decider for $A_{\text{dec}}$.

Construct $S$ deciding $A_{\text{dec}}$ as follows:

1. On input $\langle w \rangle$:
   1. Construct PDA $B(w)$ deciding $L(w)$.
   2. Construct grammar $G$ of $B(w)$.
   3. Run $T$ on input $\langle b \rangle$.
   4. If $T$ accepts, reject; if $T$ rejects, accept.

$S$ decides $A_{\text{dec}}$. Contradiction.

**Mapping Reducibility**

**Definition (Computable Function):** A function $f : \mathbb{Z}^n \rightarrow \mathbb{Z}^m$ is a computable function if

- some Turing machine $M_f$ on every input $w$, halts with just $f(w)$ on its tape.

  - e.g. Add: $\langle w \rangle \rightarrow \langle w + v \rangle$
    - Multiply: $\langle w \rangle \rightarrow \langle w \times v \rangle$
    - Dec: $\langle w \rangle \rightarrow <w> \rightarrow \langle w \rangle$

**Definition (Mapping reducible):** Language $A$ is mapping reducible to language $B$, written $A \leq_m B$, if there is a computable function $f : \mathbb{Z}^n \rightarrow \mathbb{Z}^m$, such that for every $w$:

\[
\text{Let } A \iff f(w) \in B
\]

The function $f$ is called the reduction of $A$ to $B$.

Mapping reducibility is also called as "many-to-one reducibility."
Then If $A \subseteq B$ then $(B \in \mathsf{decidable}) \Rightarrow (A \in \mathsf{decidable})$

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

Decide $M$ for $A$:

1. $N =$ "An input $u$:
   1. Compress $(u, f(u))$.
   2. Run $M$ on input and output whatever $M$ outputs."

Contrary: If $A \subseteq B$ and $A$ is undecidable, then $B$ is undecidable.

Example: 8) $A_{TM} \in \mathsf{NLOG}$

$F =$ "An input $(w)$:
1. If input is not in some form of repeat the input.
2. Construct the following machine $M$:

$M =$ "An input $x$:
1. Run $M$ on $x$.
2. If $M$ accepts accept.
3. If $M$ rejects, go to a loop."

1. Output $(x, y)$.

b) $E_{TM} \in \mathsf{log}$.

\[ f(x) \rightarrow (x, y) \text{ when $M$ rejects all inputs.} \]

g) $A_{TM} \in \overline{E_{TM}}$.

\[ f(x) \rightarrow (M, y) \text{ (as in the proof earlier).} \]

Then: If $A \subseteq B$ and $B$ is Turing-recognizable, then $A$ is Turing-recognizable.

Contrary: If $A \subseteq B$ and $A$ is not Turing-recognizable, then $B$ is not Turing-recognizable.

Then: $A_{TM}$ is unimplies $A \subseteq B$.

Then: $E_{TM}$ is neither Turing-recognizable nor co-Turing-recognizable.

$D_{TM}$:

$A_{TM} \in \overline{E_{TM}}$.

$F =$ "An input $(w, u)$:

1. Construct following $M_1$, $M_2$, and $M_3$:

$M_1 =$ "An input $x$:
1. Repeat $w$.

$M_2 =$ "An input $x$:
1. Run $M_1$ on $x$, if it accepts accept.

2. Output $(M_1, M_2)$.

$A_{TM} \in \overline{E_{TM}}$.

$G =$ "An input $(w, u)$:
Correspondence Problem

The union of PCP is a collection of strings:

\[ P = \{ [x_1], [x_2], [x_3], \ldots, [x_n] \} \]

A match is a sequence \( t_1, t_2, \ldots, t_n \) such that \( t_1, t_2, \ldots, t_n \) is a prefix.

Example:

\[ P = \{ [a], [ab], [abc], [abcd] \} \]

A match \( [a] [ab] [abc] [abcd] \)

The string \( \text{abcabcabc} \) is a match.

PCP = \{ \( <p> \): P is a collection of strings with a match \}

Theorem: PCP is undecidable.

Proof:

We show reduction from ATM via accepting computation history.

Fix any TM \( M \) and string \( w \). Let \( w \) be without infinite \( \delta \) of \( \langle M \rangle \) where \( w \) is in an accepting computation history for \( M \) on \( w \).

Formal description:

1. \( M \) or \( \delta \) never attempts to move its head off the left-hand end of the tape.
2. \( \delta \) is \( \epsilon \), in which case, \( \delta \) is in the continuation of \( P \).
3. We require that the first symbol will be \( \delta \). \( \delta \) will be in the continuation of \( P \).

\[ P(C) = \{ <\delta>: P \text{ is a collection of strings with a match that starts with the first symbol} \} \]

We first prove \( \text{ATM} \in \text{PCP} \).

Let \( M = (Q, \Sigma, \Gamma, \delta, s, \text{accept}, \text{reject}) \). Let \( w \in \Sigma^* \).

Part 1: For \( \frac{\frac{\Gamma}{\delta(0, s)}}{\text{it is a first symbol in } P} \).

Part 2: For all \( s \in \Gamma \) and any \( q \in Q \) where \( q \neq \text{reject} \),

\[ \psi(\delta(q)) = (q, \delta(q)), \text{ put } [\frac{\gamma}{\delta(q)}] \text{ into } P. \]

Part 3: For all \( q \in Q \) and any \( q \in Q \) where \( q \neq \text{ reject} \),

\[ \psi(\delta(q)) = (q, q), \text{ put } [\frac{\gamma}{\delta(q)}] \text{ into } P. \]

Part 4: For all \( q \in Q \), put \( [\frac{\gamma}{\delta(q)}] \text{ into } P. \)

Part 5: Put \( \frac{\frac{\Gamma}{\delta(0, s)}}{\text{it is a first symbol in } P} \).

TOC Page 6
Part 5: Put \( \begin{bmatrix} \frac{3}{5} \\ \frac{4}{5} \end{bmatrix} \) and \( \begin{bmatrix} \frac{4}{5} \\ \frac{3}{5} \end{bmatrix} \) into \( P \).

Part 6: For any \( a \in R \), let
\[
\begin{bmatrix} \frac{a}{5} \\ \frac{1-a}{5} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \frac{1-a}{5} \\ \frac{a}{5} \end{bmatrix}
\]
into \( P \).

Part 7: Add \( \begin{bmatrix} \frac{3}{5} \\ \frac{4}{5} \end{bmatrix} \) into \( P \).

Main beginning, let \( i = 0 \).

\[
\begin{bmatrix}
0 & 1 & 0 & 0 & 0
\end{bmatrix}
\]

Let \( d(q_0, x) = (3, 2, 2) \), so we have diagonal \( \begin{bmatrix} \frac{3}{2} \end{bmatrix} \) in \( P \).

We also have \( \begin{bmatrix} 0 \end{bmatrix} \) and \( \begin{bmatrix} \frac{1}{2} \end{bmatrix} \) in \( P \).

\[
\begin{bmatrix}
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

Let \( d(q_0, 3) = (3, 2, 2) \); we have \( \begin{bmatrix} \frac{3}{2} \end{bmatrix} \) in \( P \).

\[
\begin{bmatrix}
1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

Suppose we reach \( q_0 \).

\[
\begin{bmatrix}
2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

Considering instance \( P \) of \( (\ldots, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots) \) of \( P \).

For \( i = 0 \), \( u_i = u_i \), let
\[
\begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

Let \( P \), \( \begin{bmatrix} \frac{2}{5} \\ \frac{3}{5} \end{bmatrix} \) to form directly with \( \begin{bmatrix} \frac{2}{5} \end{bmatrix} \).

Thus let \( P \), \( \begin{bmatrix} \frac{2}{5} \\ \frac{3}{5} \end{bmatrix} \), \( \begin{bmatrix} \frac{3}{5} \\ \frac{2}{5} \end{bmatrix} \), \( \begin{bmatrix} \frac{1}{5} \\ \frac{4}{5} \end{bmatrix} \), \( \begin{bmatrix} \frac{4}{5} \\ \frac{1}{5} \end{bmatrix} \), \( \begin{bmatrix} \frac{1}{5} \\ \frac{4}{5} \end{bmatrix} \), \( \begin{bmatrix} \frac{4}{5} \\ \frac{1}{5} \end{bmatrix} \).