Turing machines codes

- May assume
  \[ M = (Q, \{0, 1\}, \{0, 1, \#\}, \delta, q_1, \#, \{q_2\}) \]

- Unary encoding
  \[
  0 \mapsto 0, 1 \mapsto 00, \# \mapsto 000, L \mapsto 0, R \mapsto 00
  \]

- \( \delta(q_i, X) = (q_j, Y, R) \) encoded by
  \[
  0^{i+1}0\ldots 010^{j+1}0\ldots 0 \rightarrow \begin{array}{c}
  X \\
  Y \\
  R
  \end{array}
  \]

- \( \delta \) encoded by
  \[
  111 \text{ code}_1, 11 \text{ code}_2 11 \ldots 11 \text{ code}_r, 111
  \]

- Encoding of Turing machine \( M \) denoted by \( \langle M \rangle \).

Numbering of Turing machines

- Lemma. There exists a Turing machine that generates the natural numbers in binary encoding.

- Lemma. There exists a Turing machine \( \text{Gen} \) that generates the binary encodings of all Turing machines.

- Proposition. The language of Turing machine codes is recursive.

- Corollary. There exist a bijection between the set of natural numbers, Turing machine codes and Turing machines.

\[
\begin{array}{c}
M \\
\rightarrow \\
\langle M \rangle \\
\rightarrow \\
\text{Equality test} + \text{counter} \\
\rightarrow \text{number } n
\end{array}
\]

Diagonalization

- Let \( w_i \) be the \( i \)-th word in \( \{0, 1\}^* \) and \( M_j \) the \( j \)-th Turing machine.

- Table \( T \) with \( t_{ij} = \begin{cases} 
1, & \text{if } w_i \in L(M_j) \\
0, & \text{if } w_i \notin L(M_j)
\end{cases} \)

\[
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
1 & 0 & 1 & 1 & 0 \\
2 & 1 & 1 & 0 & 1 \\
3 & \vdots & \vdots & \vdots & \vdots
\end{array}
\]

- Diagonal language \( L_d = \{ w_i \in \{0, 1\}^* \mid w_i \notin L(M_i) \} \).

- Theorem. \( L_d \) is not recursively enumerable.

- Proof: Suppose \( L_d = L(M_k) \), for some \( k \in \mathbb{N} \). Then
  \[
  w_k \in L_d \iff w_k \notin L(M_k),
  \]
  contradicting \( L_d = L(M_k) \).
Universal language

- \( \langle M, w \rangle \): encoding \( \langle M \rangle \) of \( M \) concatenated with \( w \in \{0, 1\}^* \).

- **Universal language**
  \[
  L_u = \{ \langle M, w \rangle \mid M \text{ accepts } w \}
  \]

- **Theorem.** \( L_u \) is recursively enumerable.
- A Turing machine \( U \) accepting \( L_u \) is called a universal Turing machine.
- **Theorem (Turing 1936).** \( L_u \) is not recursive.

Decision problems

- Decision problems are problems with answer either yes or no.
- Associate with a language \( L \subseteq \Sigma^* \) the decision problem \( D_L \)
  
  Input: \( w \in \Sigma^* \)
  
  Output: \[
  \begin{cases}
  \text{yes}, & \text{if } w \in L \\
  \text{no}, & \text{if } w \notin L
  \end{cases}
  \]
  and vice versa.

- \( D_L \) is decidable (resp. semi-decidable) if \( L \) is recursive (resp. recursively enumerable).
- \( D_L \) is undecidable if \( L \) is not recursive.

Reductions

- A many-one reduction of \( L_1 \subseteq \Sigma_1^* \) to \( L_2 \subseteq \Sigma_2^* \) is a computable function \( f : \Sigma_1^* \to \Sigma_2^* \) with \( w \in L_1 \iff f(w) \in L_2 \).

- **Proposition.** If \( L_1 \) is many-one reducible to \( L_2 \), then
  1. \( L_1 \) is decidable if \( L_2 \) is decidable.
  2. \( L_2 \) is undecidable if \( L_1 \) is undecidable.

Post’s correspondence problem

- Given pairs of words
  \[
  (v_1, w_1), (v_2, w_2), \ldots, (v_k, w_k)
  \]
  over an alphabet \( \Sigma \), does there exist a sequence of integers \( i_1, \ldots, i_m, m \geq 1 \), such that
  \[
  v_{i_1}, \ldots, v_{i_m} = w_{i_1}, \ldots, w_{i_m}.
  \]

- **Example**

<table>
<thead>
<tr>
<th>( i )</th>
<th>( v_i )</th>
<th>( w_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>111</td>
</tr>
<tr>
<td>2</td>
<td>10111</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>

\[
\Rightarrow v_2 v_1 v_3 = w_2 w_1 w_3 = 101111110
\]

- **Theorem (Post 1946).** Post’s correspondence problem is undecidable.