Background: Definitions

1. Decision problems & languages:
   - alphabet $\Sigma$ (e.g., $\Sigma = \{0, 1\}$)
   - $\Sigma^* = \{ \text{finite strings over } \Sigma \}$
     - $\{0, 1\}^*$ = all finite binary strings
   - A language $L \subseteq \Sigma^*$:
     (e.g., $L = \{ w \in \{0, 1\}^* \text{ s.t. } w \text{ contains an even number of } 1's \}$)
   - A decision problem for a language $L \subseteq \Sigma^*$:
     - Input: $x \in \Sigma^*$
     - Decide: Is $x \in L$? (Yes/No)

   $f_L : \Sigma^* \rightarrow \{0, 1\}$

   Ex: Given a graph $G=(V,E)$, is $G$ connected?

   Need to agree on some encoding scheme for graphs (say, using binary strings).

   Then
   $f_{\text{conn}} : \{0, 1\}^* \rightarrow \{0, 1\}$
   is s.t. $\forall x \in \{0, 1\}^*$,
   $f_{\text{conn}}(x) = 1$ if the graph encoded by $x$
\[ f_{\text{conn}}(x) = \begin{cases} 1 & \text{if the graph encoded by } x \text{ is connected} \\ 0 & \text{otherwise} \end{cases} \]

2. Search problems

\underline{Given:} input
\underline{Find:} output

\begin{itemize}
  \item Given a graph \( G = (V,E) \), st \( V \)
    
    Find a shortest path in \( G \)
    
    \text{from } s \text{ to } t \text{ (if it exists)}.
  
  \item Given a number \( N \),
    
    Find its prime factor (if \( N \) is composite).
\end{itemize}

3. Often, a given search problem has a corresponding decision problem such that the search problem can be "reduced" to solving the decision problem. (will see examples later)

a. Turing machine (TM)

1930's: Human Computers

\[ \Sigma = \{ 0, 1 \} \]
In each step:
- scan the current sheet
- update it
  (based on her internal state)
- then move to the sheet on your left or right
- & possibly change your internal state

\[ TM \quad M = (Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej}) \]
- \( Q \) = set of states
- \( \Sigma \) = input alphabet
- \( \Gamma \) = tape alphabet
  \( \Gamma \supseteq \Sigma \)
e.g., \( \Sigma = \{0,1\} \)
\[ r = \{ 0, 1, \bar{0}, 1 \} \]

- \( q_0 \): initial state
- \( q_a \): accepting state
- \( q_r \): rejecting state

\[ S : Q \times \Gamma \rightarrow Q \times \Gamma \times \{ L, R, \cdot \} \]

**Computation of a TM**

\[ \begin{array}{c|c|c|c|c|c|c|c} \hline x_1 & x_2 & x_3 & \ldots & x_n & u \hline \end{array} \]

Input: \( x = x_1, \ldots, x_n \in \Sigma^* \)

\[ q_0 \text{ follows } S \]

Sequence of steps:

\[ S(q_0, 0) = (q_1, 1, L) \]
Possible for TM to run forever; impossible to tell if a given TM halts on a given input.

The Halting Problem!

Language decided by a TM

\[ M = (Q, \Sigma, \Gamma, S, q_0, q_{\text{acc}}, q_{\text{rej}}) \]

decides a language \( L \subseteq \Sigma^* \) if

1. \( M \) halts on every input \( x \in \Sigma^* \),
2. \( \forall x \in \Sigma^*, \ M \text{ accepts } x \iff x \in L \)

\( M \text{ enters state } q_{\text{acc}} \)
**Remark:**

1. each TM $\leftrightarrow$ natural number  
   (finite binary encoding of TM)
   
   So $\# \text{TMs} = |\mathbb{N}|$.

2. each decision problem $f: \{0,1\}^* \rightarrow \{0,1\}$
   (infinite binary string = the values of $f$ on all binary inputs)
   
   So $\# \text{decision problems} = |\text{set of infinite binary strings}|$
   
   $= |\mathbb{R}|$

3. Since $|\mathbb{R}| > |\mathbb{N}|$, we get that 3 decision problems not computable by any TM!

---

Multi-tape TMs

![Multi-tape TMs diagram]

- input tape
- work tapes
Theorem: Every k-tape TM running in time t can be simulated by a 1-tape TM in time \( O(t^2) \).

Idea:
Is there a faster way?
No!

**Palindromes:**
\[ \text{PAL} = \{ w \in \{0,1\}^* \mid w = w^R \} \]

eg: 0110, 11, 010, 0110, 11, 010

Want a 1-tape TM to decide PAL.

\[ q_0 \xrightarrow{a} q_2, q_2, \ldots, q_n \xrightarrow{a, q_0} \]

Want to know: Is \( a = a^R \) ?

q0 remembers 0.
\[(q_0, 0) \rightarrow (p_0, 0, R)\]
\[(p_0, 0) \rightarrow (p_0, 0, R)\]
\[(p_0, 1) \rightarrow (p_0, 1, R)\]
\[(p_0, \text{\_}) \rightarrow (r_0, \text{\_}, \text{\_})\]
\[(r_0, 1) \rightarrow q \text{ reject}\]
\[(r_0, 0) \rightarrow (s, \text{\_}, \text{\_})\]
\[(r_0, \text{\_}) \rightarrow q \text{ accept}\]
\[(s, 0) \rightarrow (s, 0, \text{\_})\]
\[(s, 1) \rightarrow (s, 1, \text{\_})\]
\[(s, \text{\_}) \rightarrow (q_0, \text{\_}, R)\]
\[(q_0, \text{\_}) \rightarrow q \text{ acc}\]
2-tape TMs

\[ u \mid a_1, a_2, \ldots, a_n \mid v \]

in \ lin. \ time

\[ \text{Thm: PAL on 1-tape TM requires time } \Omega(n^2) \]

\[ \text{Thm: Any k-tape TM in time } t, \ldots, \]

\[ \text{PA} \]
can be simulated by a 2-tape TM in time $O(t \cdot \log t)$.

\textit{Universal TM}

$U$, given $(M, w)$, $M$ is some TM, $w$ input, $U$ simulates $M$ on $w$ in time $O(t \cdot \log t)$.