REGULAR$_{TM} = \{ <M> \mid M \text{ is a TM and } L(M) \text{ is a regular language} \} \quad (S219)$

Thm. REGULAR$_{TM}$ is undecidable

- Create $M_2$ = “On input $x$:
  1. If $x$ has the form $0^n1^n$ (Not Regular) accept
  2. If $x$ does not have this form, run $M$ on $w$ and accept if $M$ accepts.”

- If $M$ does not accept $w$, $M_2$ only accepts a non-regular language
- If $M$ does accept $w$, $M_2$ accepts anything $\sum^*$
  $\quad \sum^*$ is a regular language

Proof: Let $R$ be decidable for REGULAR$_{TM}$. Construct $S$ to decide $A_{TM}$

- $S$ = “On input $<M, w>$:
  1. Construct the TM as described
  2. Run $R$ on input $<M_2>$
  3. If $R$ accepts, accept; If $R$ rejects, reject”

(Never run anything on $M_2$. Just construct description to feed to $R$)

Therefore, REGULAR$_{TM}$ is undecidable

Computation Histories (S221)

- Definition: An accepting computation history for TM $M$ on string $w$ is a sequence of configurations $C_1, C_2, \ldots, C_L$ where $C_1$ is the start configuration and $C_L$ is an accepting configuration of $M$ and each $C_i$ legally follows $C_{i-1}$.
- A rejecting computation history $\rightarrow C_L$ is a rejecting configuration

Linear Bounded Automata: A TM where the tape head is not permitted to move off the portion of the tape containing input.

Deciders for $A_{DFA}$, $A_{CFG}$, $E_{DFA}$, $E_{CFG}$, every CFL decided by LBA

- Lemma (S222)
  - Let $M$ be an LBA with $q$ states and $g$ symbols in the alphabet
  - $|Q| = q$ and $|\Gamma| = g$. There are exactly $q^n g^n$ ($q^n$ states and $g^n$ memory) distinct configurations of $M$ for a tape of length $n$
\( A_{LBA} = \{ <M, w> \mid M \text{ is an LBA that accepts } w \} \)

Thm. \( A_{LBA} \) is decidable

Proof Idea:

Simulate \( M \) on \( w \)

Accept if \( M \) accepts

Reject if \( M \) rejects or loops

\( E_{LBA} = \{ <M> \mid M \text{ is an LBA and } L(M) = \emptyset \} \)

Thm. \( E_{LBA} \) is undecidable

Reduce from \( A_{TM} \)

Create an LBA s.t \( L(B) \) are accepting computation histories for \( M \) on \( w \)

Input is computation history separated by #

- Construct it to act as an emulator for \( M \) on \( w \) checking history
- \( C_1#C_2\ldots#C_3 \)
  - \( C_1 \) is a valid start configuration
  - Each \( C_{i+1} \) legally follows \( C_i \)
  - \( C_1 \) is an accepting configuration

Decider for \( A_{TM} \) – Let \( R \) be a TM that decides \( E_{LBA} \)

- \( S = \text{“On input } w: \)
  1. Construct LBA \( B \) from \( M, w \)
  2. Run \( R \) on input \( <B> \)
  3. If \( R \) rejects, accept; If \( R \) accepts, reject

If \( R \) accepts \( <B> \), then \( L(B) = \emptyset \). Thus, \( M \) has no accepting computation history on \( w \) and \( M \) does not accept \( w \). Consequently, \( S \) rejects \( <M, w> \). Similarly, if \( R \) rejects \( <B> \), the language of \( B \) is nonempty. The only string that \( B \) can accept is an accepting computation history for \( M \) on \( w \). Thus, \( M \) must accept \( w \). Consequently, \( S \) accepts \( <M, w> \). Figure 5.12 illustrates LBA \( B \)

\[ \text{Figure 5.12} \]

LBA \( B \) checking a TM computation history
Homework Problem

- Post Correspondence Problem
- PCP

Use dominoes (allowing repetition) to create matching strings on top/bottom

Ex.

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

Top and bottom are both matching

PCP is undecidable

Thm. \( EQ_{TM} \) is neither recognizable nor Co-Turing recognizable