Intro to Theory of Computation

LECTURE 17

Last time
• Reductions
• Mapping reductions

Today
• Computation history method

Sofya Raskhodnikova
A language $L$ is Turing-recognizable $\iff L \leq_m A_{TM}$

A. Only the $\Rightarrow$ direction is true.
B. Only the $\Leftarrow$ direction is true.
C. Both directions are true.
D. Neither direction is true.
### Problems in language theory

<table>
<thead>
<tr>
<th></th>
<th>$A_{DFA}$ decidable</th>
<th>$A_{CFG}$ decidable</th>
<th>$A_{TM}$ undecidable</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_{DFA}$ decidable</td>
<td>$E_{CFG}$ decidable</td>
<td>$E_{TM}$ undecidable</td>
<td></td>
</tr>
<tr>
<td>$EQ_{DFA}$ decidable</td>
<td>$EQ_{CFG}$ ?</td>
<td>$EQ_{TM}$ undecidable</td>
<td></td>
</tr>
</tbody>
</table>

3/17/2016
Proving undecidability for languages that do not involve TM descriptions

Computation history method
A linear bounded automaton (LBA) is a TM variant that has bounded tape, with the number of tape squares equal to the size of the input.
Configurations

A configuration of an LBA is a setting of state, head position and tape contents.
Exercise

How many distinct configurations does an LBA have if it has \( q \) states, \( g \) symbols in tape alphabet, and the tape of length \( n \)?

A. \( qgn \)
B. \( q + g + n \)
C. \( qg^n \)
D. \( qng^n \)
E. None of the above.
Prove that $A_{LBA}$ is decidable

$LBA = \{ \langle B, w \rangle \mid B \text{ is an LBA that accepts string } w \}$

Idea: Given $\langle B, w \rangle$, simulate $B$ on $w$.
If it halts, we know the answer.
If it loops, we can detect because $B$ repeats a configuration.

$S = \text{``On input } \langle B, w \rangle, \text{ where } B \text{ is an LBA and } w \text{ is a string:} $

1. Simulate $B$ on $w$ for $qng^n$ steps.
2. If it accepts, accept.
3. If it rejects or does not halt, reject."
An accepting computation history for a TM $M$ on input $w$ is a sequence of configurations entered by $M$ on input $w$:

$$C_0 \# C_1 \# \ldots \# C_\ell$$

- **starting configuration**
- **accepting configuration**
Starting configuration

\[ C_0 = \langle q_0 \rangle_w \]
Accepting configuration

\[ C_\ell = \ldots \langle q_{\text{acc}} \rangle \ldots \]
Each $C_{i+1}$ legally follows from $C_i$

\[ C_i = 1000 \langle q_7 \rangle 0110 \]

\[ C_{i+1} = 1000 2 \langle q_5 \rangle 110 \]
LBAs can check computation histories of TMs

Given a TM $M$ and a string $w$, we can construct an LBA that checks whether its input is the accepting computation history of $M$ on $w$. 
Prove that $E_{LBA}$ is undecidable

$E_{LBA} = \{ \langle B \rangle | B \text{ is a LBA and } L(B) = \emptyset \}$

Proof: Suppose to the contrary that TM $R$ decides $E_{LBA}$. We construct TM $S$ that decides $A_{TM}$.

$S = \text{`` On input } \langle M, w \rangle \text{, where } M \text{ is a TM and } w \text{ is a string:} \text{``}$

1. Construct an LBA $B$ from $M$ and $w$:

   $B = \text{`` On input } x, \text{ Accept if } x = C_0 \# \ldots \# C_\ell \text{ is the accepting computation history of } M \text{ on } w:}$

   1. $C_0$ is the starting configuration of $M$ on $w$
   2. Each $C_{i+1}$ legally follows from $C_i$
   3. $C_\ell$ is an accepting configuration for $M$ ”

2. Run TM $R$ on input $<B>$.

3. If it rejects, accept. O.w. reject.”
\( \text{ALL}_{\text{PDA}} \) is undecidable

\[ \text{ALL}_{\text{PDA}} = \{ \langle P \rangle \mid P \text{ is a PDA that accepts all strings} \} \]

- We can use the computation history method to show that \( \text{ALL}_{\text{PDA}} \) is undecidable.
- It follows that \( \text{ALL}_{\text{CFG}} \) is undecidable.
- It follows that \( \text{EQ}_{\text{CFG}} \) is undecidable.
# Problems in language theory

<table>
<thead>
<tr>
<th>$A_{DFA}$</th>
<th>$A_{CFG}$</th>
<th>$A_{TM}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>decidable</td>
<td>decidable</td>
<td>undecidable</td>
</tr>
<tr>
<td>$E_{DFA}$</td>
<td>$E_{CFG}$</td>
<td>$E_{TM}$</td>
</tr>
<tr>
<td>decidable</td>
<td>decidable</td>
<td>undecidable</td>
</tr>
<tr>
<td>$EQ_{DFA}$</td>
<td>$EQ_{CFG}$</td>
<td>$EQ_{TM}$</td>
</tr>
<tr>
<td>decidable</td>
<td>undecidable</td>
<td>undecidable</td>
</tr>
</tbody>
</table>
Post Correspondence Problem

Domino: \[ \begin{array}{c}
a \\
ab 
\end{array} \]. Top and bottom are strings.

Input: collection of dominos.
\[
\begin{array}{c}
\frac{aa}{aba}, \frac{ab}{aba}, \frac{ba}{aa}, \frac{abab}{b}
\end{array}
\]

Match: list of some of the input dominos (repetitions allowed) with top = bottom
\[
\begin{array}{c}
\frac{ab}{aba}, \frac{aa}{aba}, \frac{ba}{aa}, \frac{aa}{aba}, \frac{abab}{b}
\end{array}
\]

Problem: determine if a match exists.
\text{POST}\{\text{input with a match}\} \text{ is undecidable.}
Exercise

A two-dimensional automaton (2DIM-DFA) takes an \( m \times n \) rectangle as input, for any \( m, n \geq 2 \). The boundary squares contain \( \# \); internal squares contain symbols from alphabet \( \Sigma \). The transition function \( \delta : Q \times (\Sigma \cup \{\#\}) \to Q \times \{L, R, U, D\} \) indicates the next state and head movement (left, right, up, down). How many distinct configurations does a 2DIM-DFA have on a given input?

A. \((m - 2)(n - 2)|Q|\)

B. \(mn|Q|\)

C. \(mn|Q| \cdot |\Sigma|\)

D. \(mn|Q| \cdot |\Sigma|^{(m-2)(n-2)}\)

E. None of the above.
Exercise

Let $A_{2\text{DIM-DFA}} = \{ \langle D, x \rangle \mid D \text{ is a } 2\text{DIM-DFA and } D \text{ accepts } x \}$

1. Can a 2DIM-DFA loop?
2. Is $A_{2\text{DIM-DFA}}$ decidable?

A. YES to both.
B. NO to both.
C. YES to 1, NO to 2.
D. NO to 1, YES to 2.