Intro to Theory of Computation

LECTURE 17
Last time
• Countable/uncountable sets.
• Diagonalization
• Undecidable/unrecognizable languages ($A_{TM}$ is undecidable)

Today
• Reductions

Sofya Raskhodnikova

3/15/2016
Classes of languages

- **Regular**
- **CFL**
- **Recognizable**
- **Decidable**

**$A_{TM}$**

- $\{0^n1^n0^n \mid n \geq 0\}$
- $\{0^n1^n \mid n \geq 0\}$
- $1^*$

$A_{TM} = \{ \langle M, w \rangle \mid M$ is a TM, $w$ is a string, and $M$ accepts $w \}$
The fact that $A_{TM}$ is undecidable means that

A. if we are given input $<M,w>$, then $M$ is not a decider

B. there is no TM $S$ such that $L(S) = A_{TM}$

C. there is no TM $S$ that accepts on strings in $A_{TM}$ and halts and rejects on strings on strings not in $A_{TM}$.

D. Both B and C are correct.

E. None of the above.
Theorem. Language $L$ is decidable iff $L$ and $\overline{L}$ are Turing-recognizable.

Proof:
1) $L$ is decidable $\Rightarrow L$ and $\overline{L}$ are Turing-recognizable.

(Proof on the board)

2) $L$ and $\overline{L}$ are Turing-recognizable $\Rightarrow L$ is decidable.

(Proof on the board)
Prove that the following languages are Turing-recognizable

\[ A_{TM} = \{ \langle M, w \rangle | M \text{ is a TM that accepts string } w \} \]

\[ \text{HALT}_{TM} = \{ \langle M, w \rangle | M \text{ is a TM that halts on string } w \} \]

Corollary. \[ \overline{A_{TM}} \] is not Turing-recognizable.

(on the board)
Classes of languages

- Regular
- Context-Free Languages (CFL)
- Recognizable
- Decidable

\[ A_{TM} = \{ \langle M, w \rangle \mid M \text{ is a TM, } w \text{ is a string, and } M \text{ accepts } w \} \]
Problems in language theory

<table>
<thead>
<tr>
<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>$E_{TM}$ ?</td>
</tr>
<tr>
<td>$EQ_{DFA}$ decidable</td>
<td>$EQ_{CFG}$ ?</td>
<td>$EQ_{TM}$ ?</td>
</tr>
</tbody>
</table>
Now that we have an undecidable language, can we get more?

Reductions
A reduction from problem A to problem B is an algorithm for problem A that uses a subroutine for problem B.

Many of the deciders we constructed use reductions to one of these problems: \( A_{DFA}, E_{DFA}, EQ_{DFA}, A_{CFG}, E_{CFG} \).
Old Theorem. \( \text{EQ}_{\text{DFA}} \) is decidable.

\[ \text{EQ}_{\text{DFA}} = \{ \langle D_1, D_2 \rangle | D_1, D_2 \text{ are DFAs } \& \ L(D_1) = L(D_2) \} \]

Proof: The following TM \( M \) decides \( \text{EQ}_{\text{DFA}} \).

\( M = \`` \text{On input } \langle D_1, D_2 \rangle, \text{ where } D_1, D_2 \text{ are DFAs:} \`

1. Construct a DFA \( D \) that recognizes the set difference of \( L(D_1) \) and \( L(D_2) \).
2. Run the decider for \( \text{E}_{\text{DFA}} \) on \( <D> \).
3. If it accepts, accept. O.w. reject.""
How to tell a difference between a CS student and a CE student?

1. Put an empty kettle in the middle of the kitchen floor and ask your subjects to boil some water.
   - Both subjects will fill the kettle with water, turn on the stove and turn the flame on.

2. Put the kettle full of water on the stove and ask the subjects to boil the water.
   - CE student will turn the flame on.
   - CS student will empty the kettle and put it in the middle of the kitchen floor... thereby reducing the problem to the one that has already been solved!
We want to prove that language L is undecidable.

Idea: Use a proof by contradiction.
1. Suppose to the contrary that L is decidable.
2. Use a decider for L as a subroutine to construct a decider for A_{TM}.
3. But A_{TM} is undecidable. Contradiction!
Prove that $\text{HALT}_{TM}$ is undecidable

$\text{HALT}_{TM} = \{ \langle M, w \rangle \mid M \text{ is a TM that halts on string } w \}$

Proof: Suppose to the contrary that $\text{HALT}_{TM}$ is decidable, and let $R$ be a TM that decides it.

We construct TM $S$ that decides $A_{TM}$.

$S = \text{``On input } \langle M, w \rangle, \text{ where } M \text{ is a TM and } w \text{ is a string:} \n
1. \text{ Run TM } R \text{ on input } \langle M, w \rangle. \\
2. \text{ If it rejects, reject.} \\
3. \text{ If } R \text{ accepts, simulate } M \text{ on } w. \\
4. \text{ If it accepts, accept. O.w. reject.}''$

$M$ must halt on $w$, since $R$ accepted.

That’s a reduction from $A_{TM}$ to $\text{HALT}_{TM}$.
Prove that $E_{TM}$ is undecidable

\[ E_{TM} = \{ \langle M \rangle \mid M \text{ is a TM and } L(M) = \emptyset \} \]

Proof: Suppose to the contrary that $E_{TM}$ is decidable, and let $R$ be a TM that decides it. 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:}"

1. Run TM $R$ on input ???
Prove that $E_{TM}$ is undecidable

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

Proof: Suppose to the contrary that $E_{TM}$ is decidable, and let $R$ be a TM that decides it. 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:}$$

1. Construct TM $M'$. 

   $M' = \text{``On input } x, \text{ 1. Ignore the input.} \text{ 2. Run TM } M \text{ on input } w. \text{ 3. If it accepts, accept.''}$

2. Run TM $R$ on input $\langle M' \rangle$.

3. If it rejects, accept. O.w. reject.”