CS 154

Lecture 10:
Lots of Reductions,
Rice’s Theorem
Next Tuesday (2/17)

Your Midterm: At 12:50pm,
in Bishop Auditorium
(see website for more information)
Today: instead of a new homework,
you’ll get a practice midterm
Don’t panic!
Practice midterm will be harder than midterm
Next Tuesday (2/17)

Your Midterm: At 12:50pm, in Bishop Auditorium

FAQ: What is fair game for the midterm?
Everything up to this lecture

FAQ: Can I bring notes?
Yes, one single-sided sheet of notes, letter paper
Definition: A decidable predicate \( R(x,y) \) is a proposition about the input strings \( x \) and \( y \), such that some TM \( M \) implements \( R \). That is,
for all \( x, y \), \( R(x,y) \) is TRUE \( \Rightarrow \) \( M(x,y) \) accepts
\( R(x,y) \) is FALSE \( \Rightarrow \) \( M(x,y) \) rejects

Can think of \( R \) as a function from \( \Sigma^* \times \Sigma^* \rightarrow \{T,F\} \)

EXAMPLES: \( R(x,y) = \) “\( xy \) has at most 100 zeroes”
\( R(N,y) = \) “TM \( N \) halts on \( y \) in at most 99 steps”

Proposition: \( A \) is decidable if and only if there is some decidable predicate \( R \) such that
\( A = \{x \mid R(x,\varepsilon)\} \)
Theorem: A language $A$ is \textit{recognizable} if and only if there is a decidable predicate $R(x, y)$ such that:

$$A = \{ x \mid \exists y \ R(x, y) \}$$

Proof: (1) If $A = \{ x \mid \exists y \ R(x,y) \}$ then $A$ is recognizable

A TM can enumerate over all $y$’s and try them in $R$. If there is a $y$ s.t. $R(x,y)$ accepts, it will be found

(2) If $A$ is recognizable, then $A = \{ x \mid \exists y \ R(x,y) \}$

Let $M$ recognize $A$. Let $R(x,y)$ be TRUE \iff $M$ accepts $x$ in $|y|$ steps.

$$M \text{ accepts } x \iff \exists y \ R(x,y)$$
Mapping Reductions

$f : \Sigma^* \rightarrow \Sigma^*$ is a computable function if there is a Turing machine $M$ that halts with just $f(w)$ written on its tape, for every input $w$

A language $A$ is mapping reducible to language $B$, written as $A \leq_m B$, if there is a computable $f : \Sigma^* \rightarrow \Sigma^*$ such that for every $w$,

$$w \in A \iff f(w) \in B$$

$f$ is called a mapping reduction (or many-one reduction) from $A$ to $B$
Corollary: If $A \leq_m B$ and $A$ is undecidable, then $B$ is undecidable

Theorem: If $A \leq_m B$ and $B$ is decidable, then $A$ is decidable

Corollary: If $A \leq_m B$ and $A$ is undecidable, then $B$ is undecidable

Theorem: If $A \leq_m B$ and $B$ is recognizable, then $A$ is recognizable

Corollary: If $A \leq_m B$ and $A$ is unrecognizable, then $B$ is unrecognizable
Theorem: $A_{TM} \leq_m HALT_{TM}$

$f(z) :=$ Decode $z$ into a pair $(M, w)$

Construct $M'$ with the specification:

“$M'(w) =$ Simulate $M$ on $w$.

if $M(w)$ accepts then $accept$

else $loop$ $forever$”

Output $(M', w)$

We have $z \in A_{TM} \iff (M', w) \in HALT_{TM}$
Theorem: $A_{TM} \leq_m HALT_{TM}$

Corollary: $\neg A_{TM} \leq_m \neg HALT_{TM}$

Proof?

Corollary: $\neg HALT_{TM}$ is unrecognizable!

Proof: If $\neg HALT_{TM}$ were recognizable, then $\neg A_{TM}$ would be recognizable...
Theorem: $\text{HALT}_{\text{TM}} \leq_m A_{\text{TM}}$

Proof: Define the computable function:

$$f(z) := \text{Decode } z \text{ into a pair } (M, w)$$

Construct $M'$ with the specification:

"$M'(w) = \text{Simulate } M \text{ on } w.$

If $M(w)$ halts then $\text{accept}$

else $\text{loop forever}$"

Output $(M', w)$

Observe $(M, w) \in \text{HALT}_{\text{TM}} \iff (M', w) \in A_{\text{TM}}$
Corollary: $\text{HALT}_{TM} \equiv_m A_{TM}$

Yo, T.M.! I can give you the magical power to either compute the halting problem, or the acceptance problem. Which do you want?

Wow, hm, so hard to choose...

I can’t decide!
The Emptiness Problem

\[ \text{EMPTY}_{\text{DFA}} = \{ \, M \mid M \text{ is a DFA such that } L(M) = \emptyset \} \]

Given a DFA, does it reject every input?

Theorem: \( \text{EMPTY}_{\text{DFA}} \) is decidable

Why?

\[ \text{EMPTY}_{\text{NFA}} = \{ \, M \mid M \text{ is a NFA such that } L(M) = \emptyset \} \]

\[ \text{EMPTY}_{\text{REX}} = \{ \, R \mid R \text{ is a regexp such that } L(R) = \emptyset \} \]
The Emptiness Problem for TMs

$$\text{EMPTY}_\text{TM} = \{ M \mid M \text{ is a TM such that } L(M) = \emptyset \}$$

*Given a program, does it reject every input?*

Theorem: $\text{EMPTY}_\text{TM}$ is *not recognizable*

Proof: Show that $\neg A_{\text{TM}} \leq_m \text{EMPTY}_\text{TM}$

$$f(z) := \text{Decode } z \text{ into a pair } (M, w).$$

Output a TM $M'$ with the behavior:

“$M'(x) := \begin{cases} 
\text{run } M(w) \text{ and output its answer,} \\
\text{else reject}
\end{cases}$”

$$z \in A_{\text{TM}} \iff L(M') \neq \emptyset$$

$$\iff M' \notin \text{EMPTY}_\text{TM}$$

$$\iff f(z) \notin \text{EMPTY}_\text{TM}$$
The Equivalence Problem

EQ_{TM} = \{(M, N) \mid M, N \text{ are TMs and } L(M) = L(N)\}

*Do two programs compute the same function?*

Theorem: EQ_{TM} is *unrecognizable*

Proof: Reduce EMPTY_{TM} to EQ_{TM}

Let M_{\emptyset} be a “dummy” TM with no path from start state to accept state

Define \( f(M) := (M, M_{\emptyset}) \)

\[
M \in \text{EMPTY}_{TM} \iff L(M) = L(M_{\emptyset}) = \emptyset \\
\iff (M, M_{\emptyset}) \in \text{EQ}_{TM}
\]
Problem 1

REVERSE = { M | M is a TM with the property: for all w, M(w) accepts ⇔ M(w^R) accepts }.

Decidable or not?

REVERSE is undecidable.
Given a machine $D$ for deciding the language $\text{REVERSE}$, we show how to decide $A_{TM}$:

- $M(w)$ accepts $\Rightarrow L(M_w) = \{01, 10\}$
- $M(w)$ doesn't accept $\Rightarrow L(M_w) = \{01\}$

$N$ accepts $w^R$ iff $N$ accepts $w$
Problem 2 Undecidable

\{ (M, w) \mid M \text{ is a TM that on input } w, \text{ tries to move its head past the left end of the input} \}

Problem 3 Decidable

\{ (M, w) \mid M \text{ is a TM that on input } w, \text{ moves its head left at least once, at some point} \}
Problem 2  Undecidable

\[ L' = \{ (M, w) \mid M \text{ is a TM that on input } w, \text{ tries to move its head past the left end of the input } \} \]

Proof: Reduce \( A_{TM} \) to \( L' \)

On input \((M,w)\), make a TM \( N \) that shifts \( w \) over one cell, marks a special symbol $ on the leftmost cell, then simulates \( M(w) \) on the tape.

If \( M \)'s head moves to the cell with $ but has not yet accepted, \( N \) moves the head back to the right.

If \( M \) accepts, \( N \) tries to move its head past the $.

\((M,w)\) is in \( A_{TM} \) if and only if \((N,w)\) is in \( L' \)
Problem 3  Decidable

\{ (M, w) \mid M \text{ is a TM that on input } w, \text{ moves its head left at least once, at some point} \}

On input \((M,w)\), run \(M\) on \(w\) for 
\[ |Q_M| + |w| + 1 \] steps.

Accept  If \(M\)’s head moved left at all
Reject   Otherwise

\((Why\ does\ this\ work?)\)
Rice’s Theorem

Suppose $L$ is a language that satisfies two conditions:

1. (Nontrivial) There are TMs $M_{\text{YES}}$ and $M_{\text{NO}}$, where $M_{\text{YES}} \in L$ and $M_{\text{NO}} \notin L$

2. (Semantic) For all TMs $M_1$ and $M_2$ such that $L(M_1) = L(M_2)$, $M_1 \in L$ if and only if $M_2 \in L$

Then, $L$ is undecidable.

A Huge Hammer for Undecidability!
## Examples and Non-Examples

<table>
<thead>
<tr>
<th>Semantic Properties $P(M)$</th>
<th>Not Semantic!</th>
</tr>
</thead>
<tbody>
<tr>
<td>• $M$ accepts 0</td>
<td>• $M$ halts and rejects 0</td>
</tr>
<tr>
<td>• for all $w$, $M(w)$ accepts iff $M(w^R)$ accepts</td>
<td>• $M$ tries to move its head off the left end of the tape, on input 0</td>
</tr>
<tr>
<td>• $L(M) = {0}$</td>
<td>• $M$ never moves its head left on input 0</td>
</tr>
<tr>
<td>• $L(M)$ is empty</td>
<td>• $M$ has exactly 154 states</td>
</tr>
<tr>
<td>• $L(M) = \Sigma^*$</td>
<td>• $M$ halts on all inputs</td>
</tr>
<tr>
<td>• $M$ accepts 154 strings</td>
<td></td>
</tr>
</tbody>
</table>

$L = \{M \mid P(M) \text{ is true}\}$ is undecidable
Rice’s Theorem: Any nontrivial semantic L over Turing machines is undecidable.

Proof: We’ll reduce $A_{TM}$ to the language L

Define $M_\emptyset$ to be a TM that never halts

Suppose first that $M_\emptyset \notin L$

Let $M_{YES} \in L$ (such $M_{YES}$ exists, by assumption)

Reduction from $A_{TM}$ On input $(M,w)$, output:

“$M_w(x) := \text{If } (M \text{ acc. } w) \& (M_{YES} \text{ acc. } x) \text{ then ACCEPT}$
else REJECT”

If $M_\emptyset \in L$ instead, we can reduce $\neg A_{TM}$ to L. Output:

“$M_w(x) := \text{If } (M \text{ acc. } w) \& (M_{NO} \text{ acc. } x), \text{ then ACCEPT}$
else REJECT”
The Regularity Problem for Turing Machines

\[ \text{REGULAR}_{\text{TM}} = \{ M \mid M \text{ is a TM and } L(M) \text{ is regular} \} \]

*Given a program, is it equivalent to some DFA?*

**Theorem:** \( \text{REGULAR}_{\text{TM}} \) is *not recognizable*

**Proof 1:** Show that \( \neg A_{\text{TM}} \leq_m \text{REGULAR}_{\text{TM}} \)

\[ f(z) := \text{Decode } z \text{ into } (M, w). \text{ Output a TM } M': \]

\[ \quad \text{“} M'(x) := \text{if } (x = 0^n1^n) \text{ then run } M(w) \text{ else reject”} \]

\[ z \in A_{\text{TM}} \Rightarrow f(z) = M' \text{ such that } M' \text{ accepts } \{0^n1^n\} \]

\[ z \notin A_{\text{TM}} \Rightarrow f(z) = M' \text{ such that } M' \text{ accepts nothing} \]

\[ z \notin A_{\text{TM}} \iff f(M, w) \in \text{REGULAR}_{\text{TM}} \]
The Regularity Problem for Turing Machines

\[ \text{REGULAR}_{\text{TM}} = \{ M \mid M \text{ is a TM and } L(M) \text{ is regular} \} \]

*Given a program, is it equivalent to some DFA?*

Theorem: \( \text{REGULAR}_{\text{TM}} \) is *not recognizable*

Proof 2: Use Rice’s Theorem!

\( \text{REGULAR}_{\text{TM}} \) is nontrivial:
- there’s an \( M_\emptyset \) which never halts: \( M_\emptyset \in \text{REGULAR}_{\text{TM}} \)
- there’s \( M' \) deciding \( \{0^n1^n \mid n \geq 0\} \): \( M' \notin \text{REGULAR}_{\text{TM}} \)

\( \text{REGULAR}_{\text{TM}} \) is semantic:
If \( L(M) = L(M') \) then \( L(M) \) is regular iff \( L(M') \) is regular, therefore \( M \in \text{REGULAR}_{\text{TM}} \) iff \( M' \in \text{REGULAR}_{\text{TM}} \)