Monday - Memorial Day

No class today.

Wednesday

Recall: $A$ is mapping reducible to $B$, written $A \leq_m B$, means there is a computable function $f : \Sigma^* \rightarrow \Sigma^*$ such that for all strings $x$ in $\Sigma^*$,

$$x \in A \text{ if and only if } f(x) \in B.$$ 

True or False: $\overline{A_{TM}} \leq_m \overline{HALT_{TM}}$

True or False: $HALT_{TM} \leq_m A_{TM}$.

**Theorem** (Sipser 5.28): If $A \leq_m B$ and $B$ is recognizable, then $A$ is recognizable.

**Proof:**

**Corollary:** If $A \leq_m B$ and $A$ is unrecognizable, then $B$ is unrecognizable.
Strategy:

(i) To prove that a recognizable language $R$ is undecidable, prove that $A_{TM} \leq_m R$.

(ii) To prove that a co-recognizable language $U$ is undecidable, prove that $\overline{A_{TM}} \leq_m U$, i.e. that $A_{TM} \leq_m \overline{U}$.

$E_{TM} = \{ \langle M \rangle \mid M$ is a Turing machine and $L(M) = \emptyset \}$

Example string in $E_{TM}$ is ________________. Example string not in $E_{TM}$ is ________________.

$E_{TM}$ is decidable / undecidable and recognizable / unrecognizable.

$\overline{E_{TM}}$ is decidable / undecidable and recognizable / unrecognizable.

Claim: ________________ $\leq_m \overline{E_{TM}}$.

Proof: Need computable function $F : \Sigma^* \rightarrow \Sigma^*$ such that $x \in A_{TM}$ iff $F(x) \notin E_{TM}$. Define

$$F = \text{“On input } x, \text{“}$$

1. Type-check whether $x = \langle M, w \rangle$ for some TM $M$ and string $w$. If so, move to step 2; if not, output

2. Construct the following machine $M'_x$:

3. Output $\langle M'_x \rangle$.”

Verifying correctness:

<table>
<thead>
<tr>
<th>Input string</th>
<th>Output string</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\langle M, w \rangle$ where $w \in L(M)$</td>
<td></td>
</tr>
<tr>
<td>$\langle M, w \rangle$ where $w \notin L(M)$</td>
<td></td>
</tr>
<tr>
<td>$x$ not encoding any pair of TM and string</td>
<td></td>
</tr>
</tbody>
</table>
Review: Week 9 Wednesday

Please complete the review quiz questions on Gradescope about mapping reductions.

Pre class reading for next time: Introduction to Chapter 7.
Friday

Recall: A is mapping reducible to B, written $A \leq_m B$, means there is a computable function $f : \Sigma^* \rightarrow \Sigma^*$ such that for all strings $x$ in $\Sigma^*$,

$$x \in A \text{ if and only if } f(x) \in B.$$ 

$$EQ_{TM} = \{ \langle M, M' \rangle \mid M \text{ and } M' \text{ are both Turing machines and } L(M) = L(M') \}$$

Example string in $EQ_{TM}$ is _____________ . Example string not in $EQ_{TM}$ is _____________ .

$EQ_{TM}$ is decidable / undecidable and recognizable / unrecognizable .

$\overline{EQ_{TM}}$ is decidable / undecidable and recognizable / unrecognizable .

To prove, show that _____________ $\leq_m EQ_{TM}$ and that _____________ $\leq_m \overline{EQ_{TM}}$.

Verifying correctness:

<table>
<thead>
<tr>
<th>Input string</th>
<th>Output string</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\langle M, w \rangle$ where $M$ halts on $w$</td>
<td></td>
</tr>
<tr>
<td>$\langle M, w \rangle$ where $M$ loops on $w$</td>
<td></td>
</tr>
<tr>
<td>$x$ not encoding any pair of TM and string</td>
<td></td>
</tr>
</tbody>
</table>
In practice, computers (and Turing machines) don’t have infinite tape, and we can’t afford to wait unboundedly long for an answer. “Decidable” isn’t good enough - we want “Efficiently decidable”.

For a given algorithm working on a given input, how long do we need to wait for an answer? How does the running time depend on the input in the worst-case? average-case? We expect to have to spend more time on computations with larger inputs.

A language is **recognizable** if 

A language is **decidable** if 

A language is **efficiently decidable** if 

A function is **computable** if 

A function is **efficiently computable** if 

---

Definition (Sipser 7.1): For $M$ a deterministic decider, its **running time** is the function $f : \mathbb{N} \to \mathbb{N}$ given by 

$$f(n) = \max \text{ number of steps } M \text{ takes before halting, over all inputs of length } n$$

Definition (Sipser 7.7): For each function $t(n)$, the **time complexity class** $TIME(t(n))$, is defined by 

$$TIME(t(n)) = \{ L | L \text{ is decidable by a Turing machine with running time in } O(t(n)) \}$$

An example of an element of $TIME(1)$ is 

An example of an element of $TIME(n)$ is 

Note: $TIME(1) \subseteq TIME(n) \subseteq TIME(n^2)$

Definition (Sipser 7.12) : $P$ is the class of languages that are decidable in polynomial time on a deterministic 1-tape Turing machine 

$$P = \bigcup_{k} TIME(n^k)$$

*Compare to exponential time: brute-force search.*

Theorem (Sipser 7.8): Let $t(n)$ be a function with $t(n) \geq n$. Then every $t(n)$ time deterministic multitape Turing machine has an equivalent $O(t^2(n))$ time deterministic 1-tape Turing machine.
Review: Week 9 Friday

Please complete the review quiz questions on Gradescope about complexity.

Pre class reading for next time: Skim Chapter 7.