Lecture 9

CSE 431
Intro to Theory of Computation

So far:

universal TM U — diagonalization

A_{TM} is Turing-recognizable but not decidable

$\overline{A_{TM}}$ is not Turing-recognizable

Decidable = T-rec & co-Trec

$HALT_{TM} = \{\langle M, w\rangle : \text{TM } M \text{ halts on input } w\}$

Thm $HALT_{TM}$ is undecidable

Proof sketch: Show that if $\exists$ TM R deciding $HALT_{TM}$ can build a TM S for A_{TM} impossible

S : On input $\langle M, w\rangle$

Run R on input $\langle M, w\rangle$

If R rejects then reject

If R accepts then

run U on $\langle M, w\rangle$

and output its answer

First proof we'll do another today

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

Thm E_{TM} is undecidable

Proof Sketch: Suppose there is a TM E deciding E_{TM} $\leftarrow$ (E) fails

We build a new TM F that decides A_{TM} (impossible) as follows:

F

On input $\langle M, w \rangle$:
- produce new $\langle M_w \rangle$ where TM M_w erases its input and replaces it with w.
- Run E on input $\langle M_w \rangle$
- If E accepts then reject
- If E rejects then accept

Note: $L(M_w) = \begin{cases} \emptyset, & \text{if } M \text{ does not accept } w \\ \Sigma^* & \text{if } M \text{ accepts } w \end{cases}$

(M_w behaves the same on all inputs)

Correctness:

$\langle M, w \rangle \in A_{TM}$ i.e. M accepts w $\Longleftrightarrow$ $L(M_w) \neq \emptyset$ ie $L(M_w) \notin E_{TM}$

F flipped the answer that E gives on input $\langle M_w \rangle$ so correct

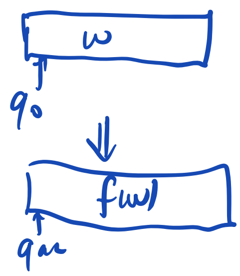
---

Each of these proofs took a (supposed) algorithm for one problem and showed how to produce an algorithm for another (in this case A_{TM})

Previously, we saw something similar when we took an algorithm for A_{DFA} and used it to solve A_{NFA} and then that one in turn was used to get an alg for A_{REG} (in that case these algorithms existed)

To define this we need to talk about TM's computing string functions

Def^n A TM M **computes a function** $f:\Sigma^* \to \Sigma^*$
iff for all $w \in \Sigma^*$ given to M as input
M halts on 1st cell of the tape
with $f(w)$ on tape followed by all blanks



Def^n We say that f is **computable** iff there is some TM that computes it

Mapping Reductions

Def^n Given $A, B \subseteq \Sigma^*$ we say that A is **mapping reducible** to B

mapping reduction

iff $\exists$ computable function $f: \Sigma^* \to \Sigma^*$
s.t. $\forall w \in \Sigma^*$

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

Notation: $A \le_m B$

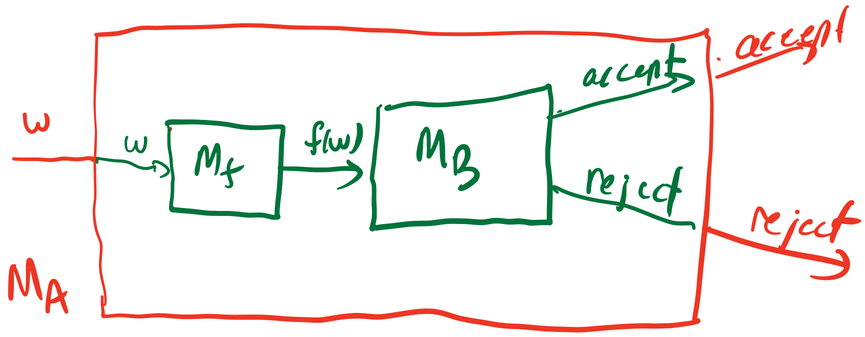
Idea: Roughly: A is (roughly) as easy as B
B is (roughly) as hard as A
Let's make this precise:

Thm If $A \le_m B$ and B is decidable then A is decidable

Proof. Since B is decidable there is a decider TM M_B for B

Since $A \le_m B$ then $\exists$ a function f computable by some TM M_f s.t. $\forall w \in \Sigma^*$ $w \in A \iff f(w) \in B$.

We build a decider M_A for A as follows



correctness follows directly:

$w \in A \Rightarrow f(w) \in B \Rightarrow M_B$ accepts $f(w) \Rightarrow M_A$ accepts w

$w \notin A \Rightarrow f(w) \notin B \Rightarrow M_B$ rejects $f(w) \Rightarrow M_A$ rejects $f(w)$

∎

<u>Thm</u> If $A \le_m B$ and B is T-rec then A is T-rec

<u>Proof</u> The same construction as the case for decidability

Proof of correctness is the same except we replace "reject" by "doesn't accept" ∎

we will use this a lot

(<u>Cor</u> If $A \le_m B$ and A is <u>not</u> decidable then B is <u>not</u> decidable

(Similarly for not T-rec)

Thm $HALT_{TM}$ is undecidable

2nd proof: Claim: $A_{TM} \leq_m HALT_{TM}$

Want: $\langle M,w\rangle \xmapsto[f]{} \langle M',w\rangle$

s.t. $\langle M,w\rangle \in A_{TM} \iff \langle M',w\rangle \in HALT_{TM}$

i.e. M accepts $w \iff M'$ halts on input w

Design for TM M': Just like M except that instead of rejecting it goes into an infinite loop

map $\langle M,w\rangle \mapsto \langle M',w\rangle$ easily computable

Correctness:
$\langle M',w\rangle \in HALT_{TM} \iff M'$ halts on input w
$\iff M'$ accepts w
$\iff M$ accepts w ∎

Example: Consider the proof that E_{TM} was undecidable we actually showed that

$$A_{TM} \leq_m \overline{E_{TM}}$$

$$\langle M,w\rangle \xmapsto{f} \langle M_w\rangle$$

∴ A_{TM} undecidable $\Rightarrow$ $\overline{E_{TM}}$ undecidable
($\Rightarrow$ E_{TM} undecidable)

This also means $\overline{A_{TM}} \le E_{TM}$

$\therefore$ E_{TM} is not Turing-recognizable

Another problem:

$EQ_{TM} = \{ \langle M_1, M_2 \rangle : M_1, M_2$ are TMs & $L(M_1) = L(M_2) \}$

Thm EQ_{TM} is undecidable

Proof Claim: $E_{TM} \le_m EQ_{TM}$

Want $\langle M \rangle \xmapsto{f} \langle M_1, M_2 \rangle$

s.t. $L(M) = \emptyset \Leftrightarrow L(M_1) = L(M_2)$

Idea: Let $M_1 = M$

and $M_2 = M_\emptyset$ a simple TM that rejects all strings

i.e. $L(M_\emptyset) = \emptyset$

$$\langle M \rangle \mapsto \langle M, M_\phi \rangle$$

computable

$\langle M \rangle \in E_{TM} \iff L(M) = \phi \iff L(M) = L(M_\phi)$

$\iff \langle M, M_\phi \rangle \in EQ_{TM}$

$\therefore E_{TM} \leq_m EQ_{TM}$

$\blacksquare$