Reducibility

Recall: A, B lang's

$A \leq_m B$ means

$\exists$ computable fcn f s.t. $\forall w$,

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

<u>Thm</u>: If $A \leq_m B$ then

1. B decidable $\Rightarrow A$ decidable
2. B T-rec $\Rightarrow A$ T-rec.

<u>Def</u> $E_{TM} := \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$

<u>Prop</u>: E_{TM} is undecidable.

<u>Proof</u>: Define <u>an m-reduction</u> f from A_{TM} to $\overline{E_{TM}}$

(thus $A_{TM} \leq_m \overline{E_{TM}}$ & so $\overline{E_{TM}}$ (and thus E_{TM}) is not decidable.

$f :=$ "On input $\underbrace{\langle M, w \rangle}_{\text{input to } A_{TM}}$, where M is a TM & w is a string:

1. Let
   $R :=$ 'On input x:
   1. Run M on input w
   [& do whatever M does on w]'
2. Output $\langle R \rangle$"

Given any input $\langle M, w \rangle$ to f,

let $\langle R \rangle := f(\langle M, w \rangle)$

(R is a TM defined within f)

<u>Case 1</u>: M accepts w

(so $\langle M, w \rangle \in A_{TM}$)

Then R accepts <u>all</u> input strings x $\therefore L(R) = \Sigma^* \neq \emptyset$

(Σ is R's input alphabet)

$\therefore \langle R \rangle \in \overline{E_{TM}}$

<u>Case 2</u>: M does not accept w

Then R does not accept <u>any</u> input string x.

$\therefore L(R) = \emptyset$

$\therefore \langle R \rangle \in E_{TM}$

$\therefore \langle R \rangle \notin \overline{E_{TM}}$

<u>Shown</u>: For all $\langle M, w \rangle$,

$\langle M, w \rangle \in A_{TM} \iff f(\langle M, w \rangle) \in \overline{E_{TM}}$

{If f's input is not of the form $\langle M, w \rangle$ for TM M & string w, then f outputs a trivial TM that rejects all inputs.} ← optional, can assume implicitly

Thus $\overline{E_{TM}}$ is undecidable by the thm last time (b/c A_{TM} is undecidable).

$\therefore E_{TM}$ is undecidable //

A direct proof that E_{TM} is undecidable. (Proof by contradiction)

Assume E_{TM} is decidable. Then let D be the following TM:

$D :=$ "On input $\langle M, w \rangle$, where M is a TM & w a string:

1. Let
   $R :=$ 'On input x:
   1. Run M on w'
2. If $\langle R \rangle \in E_{TM}$ then reject
3. Else, accept."

legitimate test b/c E_{TM} is assumed decidable.

D decides A_{TM} $\lightning$.

$\therefore E_{TM}$ is undecidable. //

<u>Def</u>: $FIN_{TM} := \{\langle M \rangle : M$ is a TM and $L(M)$ is finite$\}$

$INF_{TM} := \{\langle M \rangle : M$ is a TM & $L(M)$ is infinite$\}$

<u>Prop</u>: FIN_{TM} & INF_{TM} are undecidable.

<u>Proof</u>: $A_{TM} \leq_m INF_{TM}$ via the <u>same</u> f defined in the prev reduction: [$\langle R \rangle := f(\langle M, w \rangle)$]

$\langle M, w \rangle \in A_{TM} \Rightarrow M$ accepts w

$\Rightarrow L(R) = \Sigma^*$

$\Rightarrow L(R)$ is infinite

$\Rightarrow \langle R \rangle \in INF_{TM}$

$\langle M, w \rangle \notin A_{TM} \Rightarrow M$ does not accept $w \Rightarrow L(R) = \emptyset$

$\Rightarrow L(R)$ is finite $\Rightarrow \langle R \rangle \notin INF_{TM}$.

$\therefore A_{TM} \leq_m INF_{TM}$ $\therefore INF_{TM}$ is undecidable.

Similarly, $A_{TM} \leq_m \overline{FIN_{TM}}$ via the same function f.

Observe: $\forall$ langs A, B, if $A \leq_m B$, then $\overline{A} \leq_m \overline{B}$ (via the same m-reduction).

$\therefore \overline{A_{TM}} \leq_m FIN_{TM}$

$\therefore FIN_{TM}$ is not T-rec.

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

$\therefore E_{TM}$ is not T-rec.

Prop: $A_{TM} \leq_m FIN_{TM}$

Proof: Let

f := "On input $\langle M, w \rangle$, where M is a TM & w a string:

1. Let
   R := 'On input x:
   1. Run M on input w for $|x|$ many steps [or until M halts, whichever comes first]
   2. If M accepts w within $|x|$ many steps, then reject
   3. Else accept.'
2. Output $\langle R \rangle$"

Given $\langle M, w \rangle$ input to f, let $\langle R \rangle$ be the output $f(\langle M, w \rangle)$.

$\langle M, w \rangle \in A_{TM} \Rightarrow M$ accepts w in t steps, for some t.
$\Rightarrow R$ rejects all its input strings x such that $|x| \geq t$
$\Rightarrow L(R)$ is finite
$\Rightarrow \langle R \rangle \in FIN_{TM}$.

$\langle M, w \rangle \notin A_{TM} \Rightarrow M$ does not accept w
$\Rightarrow R$ accepts all input strings x
$\Rightarrow L(R) = \Sigma^*$
$\Rightarrow L(R)$ is infinite
$\Rightarrow \langle R \rangle \notin FIN_{TM}$

$\therefore A_{TM} \leq_m FIN_{TM}$ via f. //

Note: The same f as above m-reduces A_{TM} to INF_{TM}

$\therefore \overline{A_{TM}} \leq_m INF_{TM}$

$\therefore INF_{TM}$ is not T-rec

Summary: Neither FIN_{TM} nor INF_{TM} nor their complements, are T-rec.

Observe: A_{TM} and $\overline{E_{TM}}$ are T-rec:

$A_{TM} = L(U)$ where

U := "On input $\langle M, w \rangle$, ... :
1. Run M on input w"

$\overline{E_{TM}} = L(N)$ where

N := "On input $\langle M \rangle$ where M is a TM:
1. Cycling through all strings of the form $\langle w, t \rangle$ where w is a string and $t \geq 0$ (natural number):
   1. Run M on input w for t steps
   2. If M accepts w within t steps, then accept
   3. Else, go on to the next pair $\langle w', t' \rangle$"

For any TM M,

$\langle M \rangle \in \overline{E_{TM}} \Leftrightarrow \exists w, \exists t$, M accepts w in t steps
$\Leftrightarrow N$ accepts $\langle M \rangle$
$\Leftrightarrow \langle M \rangle \in L(N)$

$\therefore L(N) = \overline{E_{TM}}$

$\therefore \overline{E_{TM}}$ is T-rec.

TMs that construct TMs

when can this be done:

M = "On input w ... :
1. ——
⋮
Let R := 'On input x:
═══'

[Now M has $\langle R \rangle$]
⋮

If $\exists$ algo that takes $\langle w, x \rangle$ & does what R would do

This is legitimate then.