Last time:

Thm: Every infinite T-rec set has an infinite decidable subset.

Lemma: If E is an enumerator that prints strings in length monotone ascending order, then $L(E)$ is decidable.
[i.e., if E prints x & later prints w, then $|x| \le |w|$.]

Proof of the thm:
Let L be T-rec & infinite; let E be an enumerator for L. Define an enumerator E' as follows:

E' := "On no input:
1. Run E, recording all strings printed by E so far in a separate list ℓ.
2. Whenever E prints a string w, check if $|w| \ge \max\{|x| : x \text{ is in } \ell\}$. If so, print w, and add w to ℓ.
3. Continue simulating E."

Note: (1) $L(E') \subseteq L(E)$
(2) E' prints strings in length monotone ascending order, $\therefore L(E')$ is decidable by the lemma.
(3) $L(E')$ is infinite: Infinitely ~~often~~, E prints a string longer than any string it has printed so far. E' will print that string. So E' prints inf. many different strings. $\therefore$ $L(E')$ is infinite. ///

Prop: Let $f : \Sigma^* \to \mathbb{N}$ such that, for every TM M, $f(\langle M \rangle) = s \in \mathbb{N}$ such that if M accepts ε, then M does so without scanning cell s

M only these cells if M accepts ε.

[otherwise $f(\langle M \rangle)$ could be anything].

No such f is computable.

Proof: Assume otherwise: that there is such an f that is computable. Then we can decide $A_{\varepsilon,TM}$ using f as a subroutine. ↯ (shown $A_{\varepsilon,TM}$ is undecidable).
Conside the following TM

D := "On input $\langle M \rangle$ where M is a TM:
1. Let $s := f(\langle M \rangle)$
2. Run M on input ε until one of the following occurs:
   a) M accepts ε. Then accept [D accepts $\langle M \rangle$]
   b) M scans cell s. Then reject. [D rejects $\langle M \rangle$]

$D :=$ "On input $\langle M \rangle$ where M is a TM:

1. Let $s := f(\langle M \rangle)$
2. Run M on input ε until one of the following occurs:
   a) M accepts ε. Then <u>accept</u> [D accepts $\langle M \rangle$]
   b) M scans cell s. Then <u>reject.</u> [D rejects $\langle M \rangle$]

[correct by the assumption on s: If M scans cell s on input ε, then M does not accept ε.]

   c) If M rejects ε, then <u>reject</u>.
   d) If M repeats a configuration before (a,b,c) above, then <u>reject</u>.

[If M repeats a config, then M loops, not accepting ε.]

[If (a,b,c) don't happen, then (d) must happen, because there are only finitely many (depending on $\langle M \rangle$) possible configs of M, so M must repeat a config, thus loop, not accepting ε.]

[D must keep track of M's complete computation on input ε.]

D halts for all $\langle M \rangle$, and accepts iff $\langle M \rangle \in A_{\varepsilon,TM}$ [i.e., M accepts ε].

$\therefore D$ decides $A_{\varepsilon,TM}$ $\lightning$

$\therefore f$ is not computable. $\blacksquare$

<u>Reducibility</u>

<u>Def</u>: Let A & B be languages (over Σ^*). Say that A <u>mapping-reduces</u> (<u>m-reduces</u>) to B (written <u>$A \leq_m B$</u>) [m = "mapping"] if there exists a computable function $f: \Sigma^* \to \Sigma^*$ such that, for all $w \in \Sigma^*$,

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

f is called a mapping reduction (m-reduction) from A to B, sometimes written $f: A \leq_m B$.

<u>Thm</u>: Let $A, B \subseteq \Sigma^*$ and suppose $A \leq_m B$. Then:

1) If B is decidable, then A is decidable.
2) If B is T-rec, then A is T-rec.

["A is no harder than B"]

<u>Proof</u>: Let f be an m-reduction from A to B.
f is computable and $\forall w \in \Sigma^*$:

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

(1) Suppose B is decidable.
Decider for A:

$D :=$ "On input w:
1. Let $x := f(w)$.
   [ok because f is computable]
2. If $\underbrace{x \in B}_{\text{[ok because B is decidable]}}$, then <u>accept</u>;
   else <u>reject</u>."

D is a decider, and for all $w \in \Sigma^*$:

$$w \in A \iff f(w) \in B$$
$$\iff x \in B$$
$$\iff D \text{ accepts } w.$$

Thus $L(D) = A$, that is D <u>decides</u> A. //(1)

For (2):
Suppose B is T-rec.
Let M be a TM such that $B = L(M)$. Let TM

$N :=$ "On input w:
1. Let $x := f(w)$.
2. Run M on input x"
   [do what M does]

optional

For all $w \in \Sigma^*$
(f is an m-red A to B)

$$w \in A \iff x \in B$$
$$\iff M \text{ accepts } x$$
$$\iff N \text{ accepts } w$$

$\therefore A = L(N)$.
$\therefore A$ is T-rec. $//_{(2)}$ $\boxed{///}$

<u>Cor</u>: If $A \leq_m B$ and A is undecidable, then B is undecidable & if A is not T-rec, then B is not T-rec.

<u>Ex</u>: $A_{TM} \leq_m A_{\varepsilon,TM}$

[<u>Cor</u>: $A_{\varepsilon,TM}$ is undecidable]

Let f be the following function:

$f :=$ "On input $\langle M, w \rangle$ where M is a TM and w is a string:
1. Let TM
   $R :=$ 'On input x:
   a) Run M on input w'
2. Output $\langle R \rangle$"

Then as before

$$\langle M, w \rangle \in A_{TM} \iff \underbrace{f(w)}_{\langle R \rangle} \in A_{\varepsilon,TM}$$

$\therefore A_{TM} \leq_m A_{\varepsilon,TM}$.

If w is not the form $\langle M, w \rangle$ (M TM, w string), then f outputs some fixed string $x_0 \notin A_{\varepsilon,TM}$.
Can assume this tacitly from now on.

<u>Prop</u>: The $\leq_m$-relation on languages is reflexive ($A \leq_m A$) and transitive
$(A \underset{f}{\leq_m} B \;\&\; B \underset{g}{\leq_m} C \Rightarrow A \underset{g \circ f}{\leq_m} C)$.