(Hence $\le_p$ means polytime reductions.)

=

Def: L is PSPACE-complete if (1) $L \in$ PSPACE
(2) $L' \in$ PSPACE, then $L' \le_p L$.

=

[Study for final: Homework for this year + exam problems from previous years.]

=

There are usually languages that are easily seen to be complete.

CPSC 421/501

Let A be PSPACE complete.
Then

$$NP^A \subset NPSPACE \underset{\text{(Ch 8)}}{\subset} PSPACE \subset P^A$$

So $NP^A \subset P^A$. (Ch 9)

=

PSPACE-complete
NP-complete

=

Recall: Say that language, L, is NP-complete if $L \in NP$, and $L_1 \in NP$ then $L_1 \le_p L$.

Claim: NP-EASY is NP-complete

Claim: PSPACE-EASY is PSPACE-complete

=

Compare: Other NP-complete problems: SAT, 3SAT, PARTITION, 3COLOR, ..

[Compare: Other PSPACE-complete problem: TQBF, ~~certain~~ games]

=

PSPACE-EASY $\in$ PSPACE ?

Input: $\langle$ Turing machine, input, $\underbrace{1 \ldots 1}_{s}$ $\rangle$

Example: Let

NP-EASY = $\{ \langle M, w, 1^t \rangle$ s.t. M is a non-det T.m., $w \in \Sigma_M^*$ (i.e. w is a valid input to M), there is some accepting path for M on input w after time t $\}$

=

PSPACE-EASY = $\{ \langle M, w, 1^s \rangle$ s.t. M is a Turing machine, $w \in \Sigma_M^*$, M ~~accepts w~~ has an accepting path to w that takes space $\le s$ $\}$

within space

$\le \underline{\text{poly}(|\langle M\rangle| \cdot |w| \cdot s)}$

Universal TM to run M on input w and to run as long as we don't use space more than s

Remark: if $1^s \rightarrow$ binary description of s, input $\langle M, w, \text{s in binary}\rangle$ could be $\le \log_2 s + |\langle M, w\rangle|$

$\cancel{\langle}\, \{1,\dots,a\} \leftarrow$ states, $\{1,\dots,b\} \leftarrow$ input symbols, ...

$$\langle M, w, \underbrace{1 \dots 1}_{s} \rangle$$

$$|\langle M, w, \underbrace{1 \dots 1}_{s} \rangle| = n$$

$\ge$ ~~descr.~~ length of description of M

$\ge$ length of w

$\ge s$

Want to show

$\langle M, w, 1^s\rangle$ accepted by some machine/algorithm

input ~~←~~

$\log_2 a, \log_2 b, \log_2 c,$

δ-values: $\delta(1,1)\#\delta(1,2)\#$

... $\delta(a,c)\#$

w description: first letter #
second letter #
...
last letter #

1^s

$$\underbrace{1\,1\,1 \dots 1}_{s}$$

which is too short...

So here: machine M, input w, run in space 1000, input:

$$\langle M\rangle, \langle w\rangle, \underbrace{1\,1 \dots 1}_{1000}$$

so input is of size ≥ 1000

So now:

$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$

$Q = \{1,\dots,a\}$, $\Sigma = \{1,\dots,b\}$, $\Gamma = \{1,\dots,c\}$

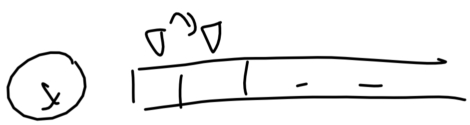
blank $= b+1 \le c$, $q_0 \leftrightarrow 1$, ...

to go from one step to another takes

time: at least linear in $\langle M \rangle$

$\underline{\text{step } i \to i+1}$: (linear in i) $(\log_2 c)$

step 2



total time in m-steps

$\leq \; |\langle M \rangle| \cdot \text{order}(m) \, (\log_2 c)$

$\cdot \, m \; + \; t \; + \; |w|$.

steps

$+$ etc. $\leq \text{poly}\left(\begin{matrix} |\langle M \rangle| \cdot \\ |w| \cdot t \end{matrix} \right)$

Rem: NP Easy similar:

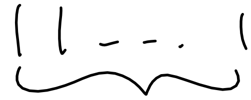
input: $\langle M, w, 1^t \rangle$

t = time to allow:

input: $\log_2 a \,\#\, \log_2 b \,\#\, \log_2 c \,\#$

$\delta(1,1) \,\#\, \ldots \,\#\, \delta(a,c) \,\#$

$\#$ first let w $\#$ 2^{nd} let w $\#$ ... $\#$ last let w



t

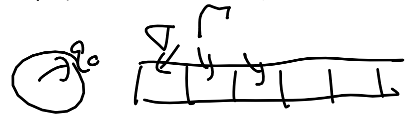
Can we accept within time t.

=

Then we write universal machines...

step 1

$\langle M \rangle$:

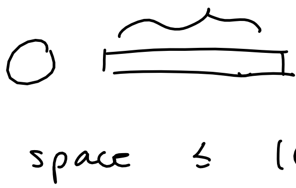


Now: $L \in NP$, $L \leq$ NP-EASY;

$L \in$ PSPACE, $L \leq$ PSPACE-EASY;

=

$L \in$ PSPACE: accepted by T.M. M in some path $\leq 10n^{20}$ space



$|w| = n$

space $\leq 10\,n^{20}$

=

Given w: tack on description of M

write w

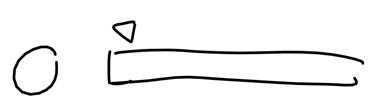
write $1 \ldots 1$ length $10\,n^{20}$

similarly: to simulate

M on input w

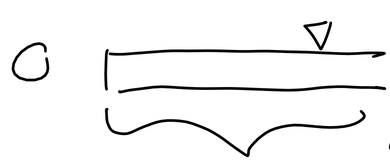
making sure not to take more than space S

$\leq \text{poly}\left(|\langle M \rangle| \cdot |w| \cdot S \right)$



step 1

⋮



S

tape head goes no farther than S

finite string

length $|w| \cdot (\log_2 b)$, or $O(|w|)$

$$w \xrightarrow{f} \langle M \rangle, \langle w \rangle,$$

$$\underbrace{1 \_ \_ \_ 1}_{\text{length } 10|w|^{20}}$$

poly in $|w|$

f is polytime