Non-deterministically guess ____ and verify that it is indeed a ____ in poly time.

(2) DOUBLE-SAT resembles SAT: probably SAT $\leq$ DOUBLE-SAT

$w \in \Sigma_{SAT} \longrightarrow f(w) \in \Sigma_{DOUBLE\text{-}SAT}$

s.t.
- f can be computed in poly time
- $w \in$ SAT $\Leftrightarrow$ $f(w) \in$ DOUBLE-SAT

Office hours today 3-5 pm
Monday x 836

8th floor "Board Room"

Sprinting through final 2014, 2011

2014, Problem 4:

DOUBLE-SAT = { <f> | with at least 2 satisfying assignments }

Show NP-complete:

Don't forget

(1) DOUBLE-SAT is in NP

5: Whole bunch of short answers:

(a) Show if $C_1, C_2, \ldots$ are countable sets then so is $C_1 \cup C_2 \cup \ldots$

$C_1 = \{ C_{11}, C_{12}, C_{13}, C_{14}, \ldots \}$

$C_2 = \{ C_{21}, C_{22}, C_{23}, \ldots \}$

$C_3 = \{ C_{31}, C_{32}, \ldots$

$C_{11}, \quad C_{12}, C_{21}, \quad C_{13}, C_{22}, C_{31}, \ldots$

any C_{ij} is in $(i+j-1)^{th}$ diagonal and reached in finite time

i.e. $w \in$ SAT $\Rightarrow$ $f(w) \in$ DOUBLE-SAT

and (don't forget)

either
- $w \notin$ SAT $\Rightarrow$ $f(w) \notin$ DOUBLE-SAT

OR
- $f(w) \in$ DOUBLE-SAT $\Rightarrow$ $w \in$ SAT

In this reduction

<formula> $\longrightarrow$ <formula + new irrelevant variable>

$x_1, x_2, \ldots, x_n$

$\varphi \longrightarrow \varphi$ AND $(x_{n+1}$ or NOT $x_{n+1})$

$k = 1, 2, 3, \ldots$

$NSPACE(n^k) \subset SPACE(n^{2k})$

$\subset PSPACE$

$NPSPACE =$

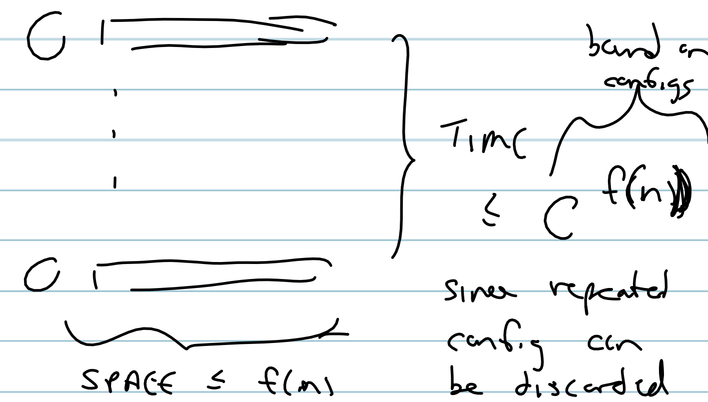
$\bigcup_k NSPACE(n^k)$

$\subset PSPACE$

(b) What does Savitch's thm say, why does it show NPSPACE = PSPACE

Savitch: $NSPACE(f(n))$

$\subset SPACE((f(n))^2)$



and f is polytime.

So on input w, run f (polytime), ask oracle if $f(w) \in A$.

=

$\left.\begin{array}{l} TIME^A(n^3) \\ NTIME^A(n^5) \\ PSPACE^A \\ NP^A \\ P^A \end{array}\right\}$ ~~oracle~~ oracle T.m. with oracle A

(c) Let A be PSPACE-complete

Is $PSPACE \subseteq P^A = \bigcup_k TIME^A(n^k)$

(same idea PSPACE replaced by NP)

$L \in PSPACE$ PSPACE-complete

$L \le_{\text{poly time}} A$

means $w \mapsto f(w)$

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

$L \in NP \Rightarrow$

$\exists$ non-det T.m, M, constants c_1, c_2 s.t.

for any w, M accepts w within time $c_1 |w|^{c_2}$.

~~$w \in NP$~~

$w \in \Sigma_L$,

$w \in L \Leftrightarrow$

$\langle M, w, 1^{c_1 |w|^{c_2}} \rangle \in L_{NP\text{-easy}}$

is poly-time reduction

In Chapter 9, the only thing we did:

there is an oracle, A, (A is just some language) for which

$$NP^A = P^A$$

=

(d) Let $L_{NP\text{-easy}}$ be

$\{ \langle M, w, 1^t \rangle \mid M$ non-det T.m. that accepts w within time $t \}$

Show that $L \in NP$ then $L \leq L_{NP\text{-easy}}$

More generally:

any A,

is there T.m. with oracle A that $\left\{ \begin{matrix} \text{decides} \\ \text{recognizes} \end{matrix} \right\}$ $HALT^A$?

If A is $\emptyset$, i.e. no oracle

=

T.m. $\longrightarrow$ T.m. all have special $\left\{ \begin{matrix} \text{oracle} \\ \text{subroutine} \\ \text{button} \end{matrix} \right\}$ A

(e) (f) Let HALT = Halting problem for Turing machines,

let

$HALT^A$ = Halting problem for T.m. with oracle A.

Is there a T.m with oracle $\overline{HALT}$ that $\left\{ \begin{matrix} \text{decides} \\ \text{recognizes} \end{matrix} \right\}$ $HALT^{HALT}$

(f) decides

(e) recognizes

without oracles:

$HALT_{TM}$ { can be recognized by universal T.m.

{ can't be decided

(P, d)

programs T.m all with the same A

(g),(h) not relevant to 2015