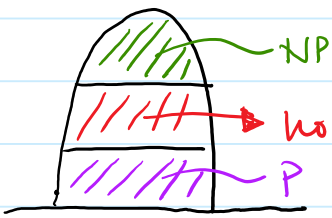
Sunday, April 15, 2018 9:19 PM

Two Theorems about the structure of NP

Ladner's Theorem: If P ≠ NP then ∃ L ∈ NP such that L ∉ P and L is not NP-complete.



Proof uses "lazy diagonalization" & is similar to proof of NTIME hierarchy.

NP Complete languages are: computationally hard. expressive

(boolean satisfiability can express any language in NP.)

A sparse language contains ≤ poly(n) strings of length ≤ n. $\forall n \geq 1$.

(out of the $2^{n+1}-1$ possible strings of length ≤ n)

It can be shown that if P ≠ NP then there is no sparse NP-complete language.

Will show a weaker version of this result:

≤ n strings of length ≤ n.

A language is unary if it is a subset of 1^*.

Sunday, April 15, 2018 9:18 PM

Theorem (Berman '78)
If a unary language is NP-Complete, then P=NP.

Proof Let U be a unary language and assume that SAT $\propto$ U by reduction R.

We will show a poly-time alg. for SAT

$\phi(x_1, \ldots, x_n)$ instance of SAT

Consider a partial assignment to the first i variables $t_1, \ldots, t_i$. This gives a new Boolean formula on n-i variables:

$$\phi(t_1, \ldots, t_i, x_{i+1}, \ldots, x_n) \Leftarrow$$

After each assignment:

$x_3 \leftarrow T$

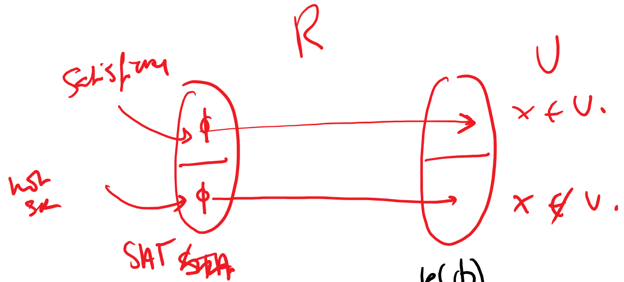
$\cancel{(x_3 \vee \neg x_5 \vee x_7)}$ If a literal becomes true, remove any clause in which it participates.

$(\cancel{\neg x_3} \vee x_9 \vee \neg x_6)$ If a literal becomes false, remove it from any clause in which it participates.

Eventually, we will get:

- An empty clause ϕ not satisfied.
- No clauses left ϕ satisfied.

Note: $\phi(t_1, \ldots, t_i, x_{i+1}, \ldots, x_n)$ is a Boolean formula to which R can be applied (size $\leq \phi(x_1 \ldots x_n)$).



$R(\phi)$ → •
$R(\phi')$ → •

Consider an exhaustive search to determine if ϕ is satisfiable

$\phi(x_1 \cdots x_n)$

$\phi(1, x_2, \ldots, x_n)$ $\phi(0, x_2, \ldots, x_n)$

⋮ ⋮

Can determine at each leaf if ϕ is satisfied.

$\phi(1,1,\cdots,1)$ T/F

$\phi(0,0,\ldots,0)$ T/F

Instead of exhaustively searching the entire tree, will use R as a hashing scheme to prune the tree.

Note: if $\underline{R}(\phi) = \underline{R}(\phi')$ then ϕ is satisfiable iff ϕ' is satisfiable.

Keep a list:

$[R(\phi(t)), Y/N]$ $[R(\phi(t')), Y/N] \ldots$

Y/N → tells if $\phi(t)$ is satisfiable.

t is a generic partial assignment.

Note $|R(\phi)| \leq \text{poly}(|\phi|)$

All the $|R(\phi(t))| \leq \text{poly}(|\phi|)$

# of strings in 1^* length $\leq \text{poly}(|\phi|)$ is $\leq \text{poly}(|\phi|)$.

Since $R(\phi(t)) \subseteq 1^*$ there can be at most $\text{poly}(|\phi|)$ distinct values. $\Rightarrow$ Size of table $\leq \text{poly}(|\phi|)$.

Mon April 16, 2018 - page 4

Sunday, April 15, 2018 9:19 PM

Eval [$\phi(t)$]

{

Compute $R(\phi(t))$ If not in 1^* then return NO.

Check if $R(\phi(t))$ is in the table.

If so, return correct answer.

If not:

Eval [$\phi(t0)$]

Eval [$\phi(t1)$]

If either is satisfiable.

Insert [$R(\phi(t))$, Y] into table

Return YES.

If neither is satisfiable

Insert [$R(\phi(t))$, N] into table.

Return NO.

}

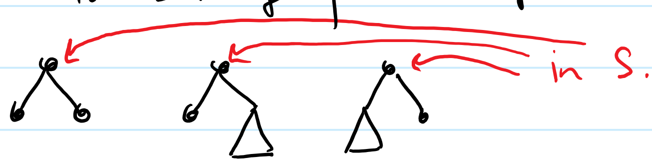
Suppose M nodes of the tree are visited

Running time: $O(M \cdot \text{poly}(|\phi|))$

Need to bound M.

Prune the tree so that it only contains visited nodes. [each node has 2 or 0 children]

Let S be the set of parents of leaves



$4|S| \geq M$

Can show by induction # interior nodes $\leq$ # leaves $- 1$

leaves $\leq 2|S|$

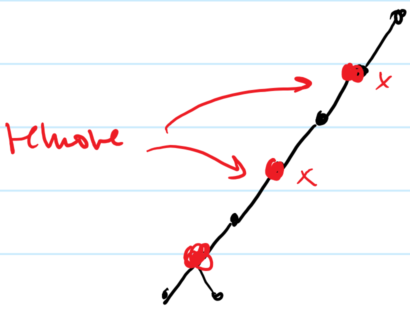
$M \leq$ int nodes $+$ # leaves

$\leq 2(\text{\# leaves}) \leq 4|S|$.

Mon April 16, 2018 - page 5

Sunday, April 15, 2018 9:19 PM

Remove from S any node which is an ancestor of another node to get a new set S'.



Each node in S can cause the removal of $\leq n$ nodes in S

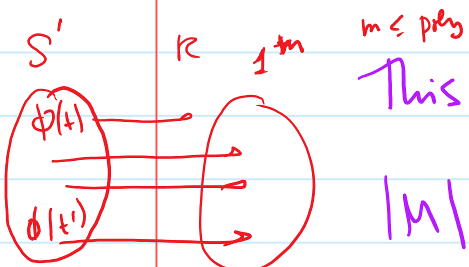
$|S| \leq n \cdot |S'|$.

The length of a path from a vertex to the root is $\leq n$.

$m \leq 4|S| \leq 4n|S'|$.

Will show that for $\phi(t)$ + $\phi(t') \in S'$

$R(\phi(t)) \neq R(\phi(t'))$.

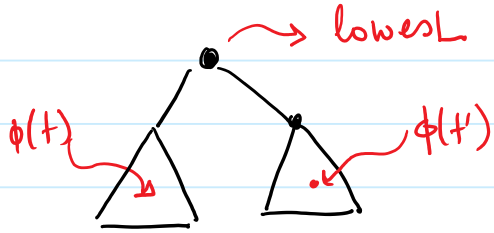


This will imply that $|S'| \leq \text{poly}(|\phi|)$.

$|M| \leq 4n|S'| \leq \text{poly}(|\phi|)$

$\phi(t)$ can not be an ancestor or descendant of $\phi(t')$.
(by the way S' was constructed)

lowest common ancestor of $\phi(t)$ + $\phi(t')$ is not in S'.



Suppose $\phi(t)$ is visited before $\phi(t')$

Call to EVAL $[\phi(t)]$ completes before $\phi(t')$ is reached.

If $R(\phi(t)) = R(\phi(t'))$ there would have been no recursive calls to the children of $\phi(t') \Rightarrow \phi(t')$ not a parent.

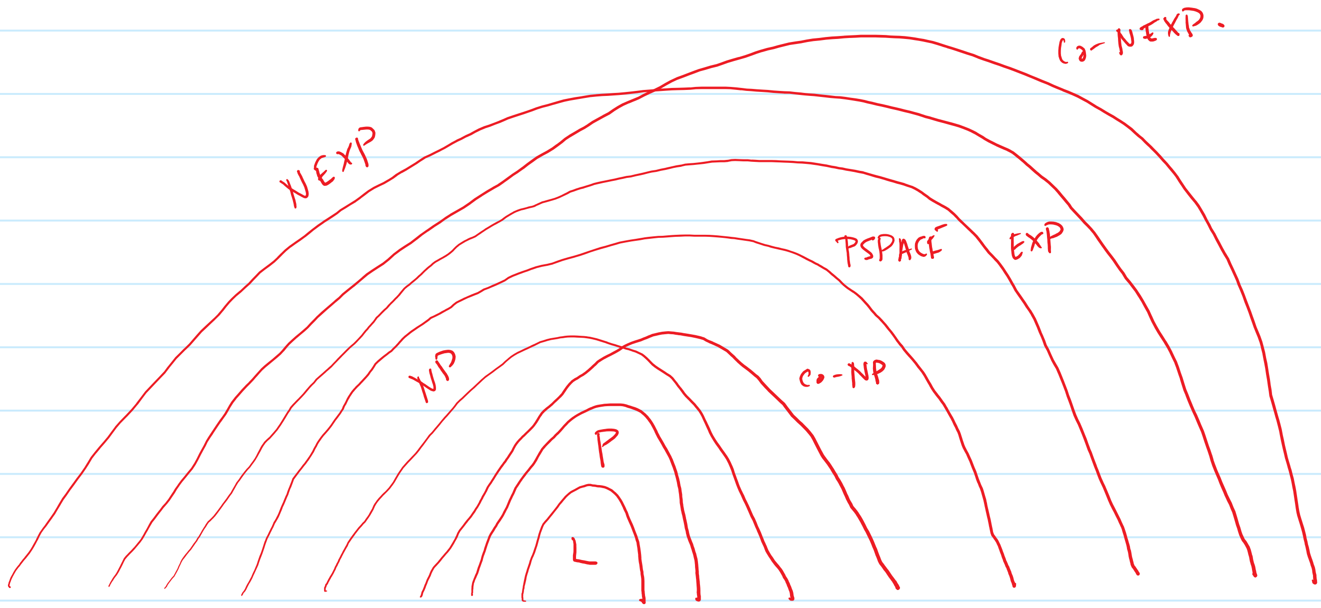
Sunday, April 15, 2018 9:19 PM

Recap on NTIME classes:
NP, co-NP, NEXP.

Open questions: $P \stackrel{?}{=} NP$ $NP \stackrel{?}{=} co\text{-}NP$

NP has intermediate problems (unless P=NP).
Sparse/unary languages not NP-C unless P=NP.

Draw diagram for: L, P, NP-co-NP, PSPACE, EXP
NEXP, co-NEXP.



Mon April 16, 2018 - page 7

Sunday, April 15, 2018 9:19 PM

Non-deterministic Space

this means that all paths must terminate.

NSPACE(f(n)) - languages decidable by a multi-tape NTM that touches $\leq f(n)$ squares of the work tape along any computational path.

Robust Non-deterministic Space classes:

$$NL = NSPACE(\log n)$$
$$NPSPACE = \bigcup_{k \geq 1} NSPACE(n^k).$$

Relationships between NTIME & NSPACE classes.

On input x ($|x| = n$) a t(n) space bounded device (det or non-det) has at most

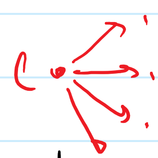
read only → $\underbrace{n \times t(n)}_{\text{head position}} \times \underbrace{|Q|}_{} \times \underbrace{\Sigma}_{\text{tape contents}}^{t(n)\cancel{k}}$ configurations

work tape.

Configuration Graph: for NTM M on input x

Vertices ⟷ Configuration. $C \to C'$

Edges ⟷ One step of M.



For deterministic M, out-degree 0 or 1.

For non-deterministic M, out-degree is bounded by a constant c.

Monday, April 16, 2018 9:05 AM

Size of Gr,x

$n \times t(n) \times |Q| \times \Sigma^{|t(n)|}$

$t = \log$ $\Rightarrow$ poly

Does NTM M accept x?

Boils down to connectivity.

Is there a path from $C_{x,start}$ to an accepting config?

Connectivity can be answered in time poly in the size of the graph.

$t(n) = \log n$ NL $\subseteq$ P.

$t(n) = n^k$ NSPACE $(n^k) \subseteq$ EXP for all k.

$\Rightarrow$ NPSPACE $\subseteq$ EXP.

This suggest a problem that could be complete for NL:

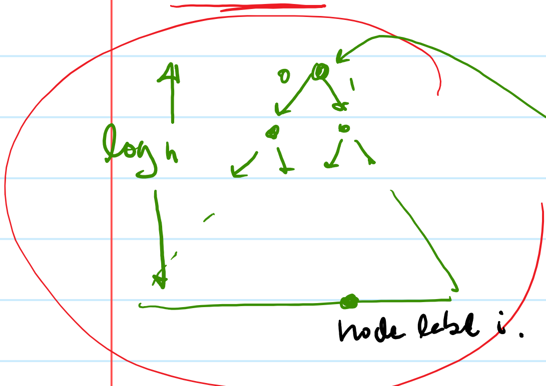
STCONN is NL-complete.

ST- Connectivity : (STCONN).

Input: $G = (V, E)$ $s, t \in V$

Is there a path from s to t in G?

STCONN $\in$ NL



$v \leftarrow s$.

clock = 0.

use non-determinism to "guess" node v'

If $(v, v') \notin E$ reject.

Else increment clock

if $v' = t$ accept.

if clock = n reject

$v \leftarrow v'$

Repeat

Monday, April 16, 2018 9:14 AM

No path from s to t?
Every computation path will hit a dead end or time will run out.

If $\exists$ path from s to t
$s = v_0 \to v_1 \to v_2 \cdots \to v_\ell = t.$
$\ell \leq n$
there is a sequence of guesses that will lead to t.

Now prove: $\forall L \in NL \quad L \propto STCONN$
log-space reduction.

There is an NTM M that uses log n cells of work tape that decides L.

Will describe logspace R:
$x \longrightarrow G_x, s, t.$

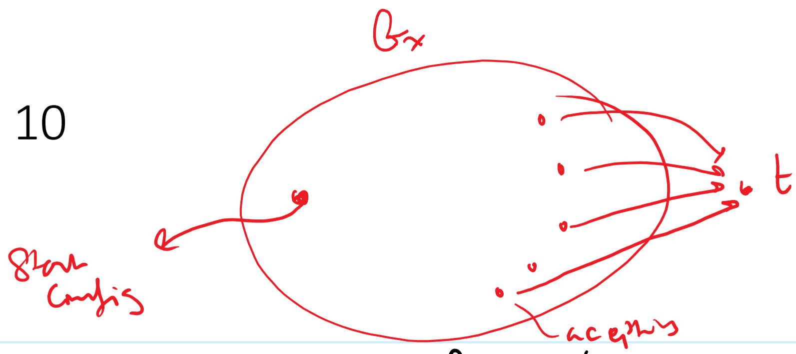
$x \in L$ iff $\exists$ path from s to t in G_x

On input x $|x| = n$. there will be
$n \times t(n) \times |Q| \times |\Sigma|^{t(n)}$ $\quad t(n) = \log.$

Each can be described w/ $\log(n \times t(n) \times |Q| \times |\Sigma|^{t(n)})$ bits.
$O(t(n))$.

Mon April 16, 2018 - page 10

Monday, April 16, 2018 9:21 AM



R: run through all pairs of configurations on the work tape.

Determine if one can be reached from the other in one step of M.
If so → output the edge.

Add a new node t
Add an edge from every accepting configuration to t.

Output: s = start config t = new node.