Finish proof that

Lecture 10

TQBF is PSPACE-complete

Last time: Established

$TQBF \in PSPACE$

Now: $\forall L \in PSPACE$

$L \leq_p TQBF$.

- M_L which decides L
- $\forall x \in \{0,1\}^n$

$M_L(x)$ decides if

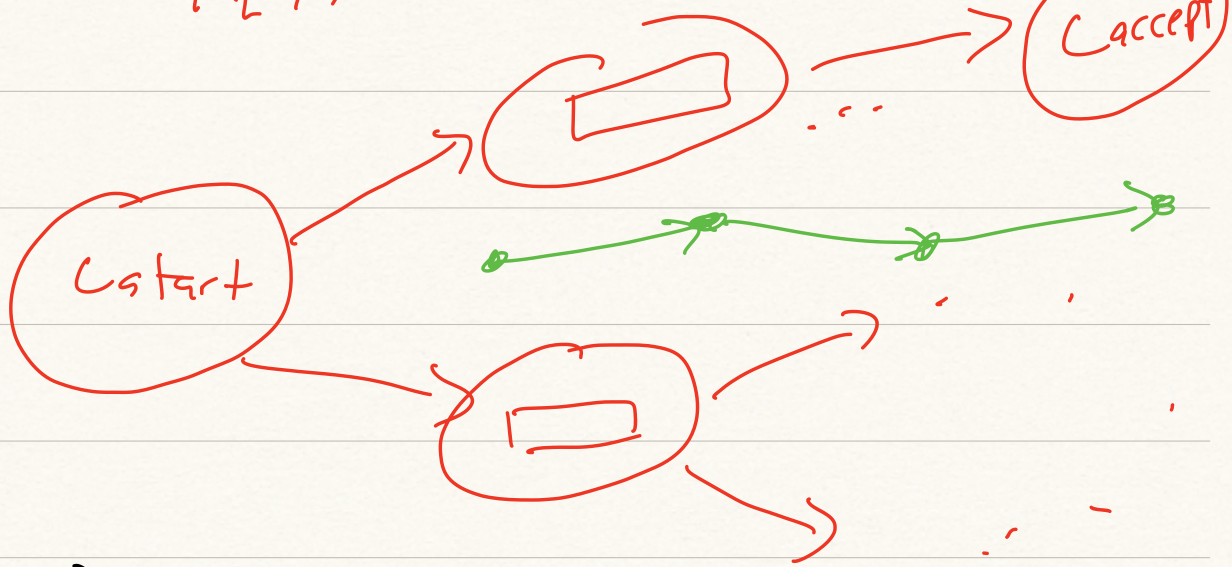
$x \in L$ in space

$O(\underbrace{s(n)})$

c n

$S(n) = n^{c}$ for some c

Given M_L, x, we can construct config. graph $G_{M_L,x}$ of $M_L(x)$



① $M_L(x) = 1 \iff$ $\exists$ path in $G_{M_L,x}$ from C_{start} to C_{accept}

② We only need $O(S(n))$ bits to encode any configuration

(3) $\exists$ formula ψ s.t.

$\psi(c, c') = 1 \iff$

$O(S(n))$

c, c' are valid configs and $c \to c'$ under M_i's trasition function (on input x)

Want: reduction f s.t.

PSPACE lang

$x \in L \iff f(x) \in TQBF$

want to define a QBF ϕ s.t.

$x \in L \iff \underline{\phi = 1} \Rightarrow \in TQBF$

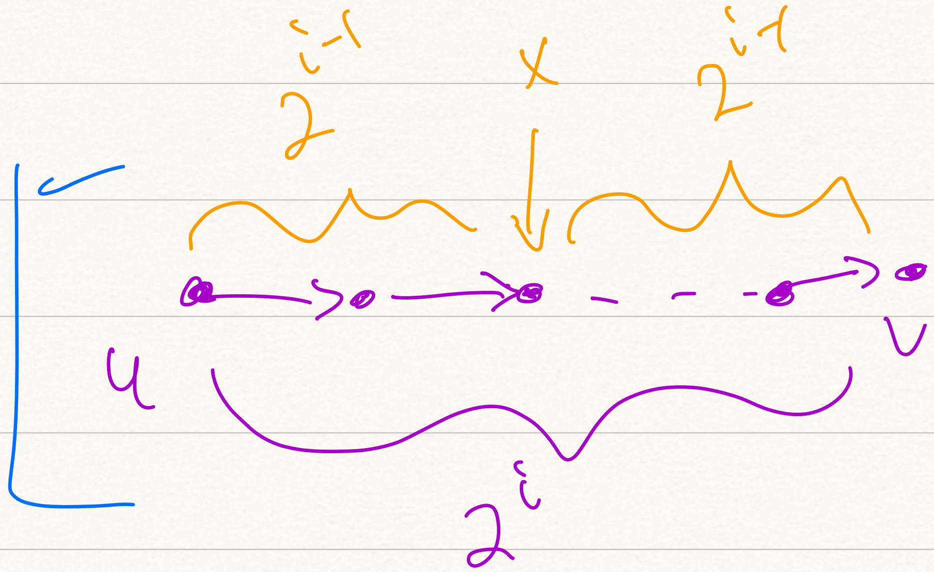
$M_L(x) = 1 \iff$

Path $\iff$
from
C_{start} to
C_{accept}
in $G_{M_L, x}$

First Attempt

- In any directed G, if there is a path from u to v of length at most 2^i

Then $\exists$ node x s.t.



path $u \to x$ of length $\le 2^{i-1}$

$x \to v$ of length $\le 2^{i-1}$

- We'll build ϕ recursively
  - <u>Base Case</u>:

$$\phi_0 = \psi$$

adjacent config formula

$\to \phi_i(c, c') = 1$ iff

$\exists$ path of length

$\le 2^i$ between

configurations c, c'

- $\phi_k = \phi$

$k = 2^{O(s(n))}$ = # of nodes in $G_{M,x}$

$\phi(C_{start}, C_{accept}) = 1$

Define

$$\phi_i(c,c') = \exists c'' \; \phi_{i-1}(c,c'') \wedge \phi_{i-1}(c'',c')$$

$O(s(n))$

ϕ_0 $\qquad |\phi_1| \geq 2 \cdot |\phi_0|$

ϕ_1 $\qquad |\phi_i| \geq 2 \cdot |\phi_{i-1}|$

$\vdots$

ϕ_k $\qquad k = O(s(n))$

$|\phi_k| \geq 2^{O(s(n))}$ $\qquad$ X

Insight: Define equivalent QBF to

$$\phi_i(c,c') = \exists c'' \; \phi_{i-1}(c,c'') \wedge \phi_{i-1}(c'',c')$$

which only increases the size by at most $O(s(n))$ space.

$$\phi_i(c,c') = \boxed{\exists c''}\ \forall d_1 \forall d_2$$

$$\left[(d_1 = c \wedge d_2 = c'') \textcircled{1} \vee (d_1 = c'' \wedge d_2 = c') \right] \textcircled{2}$$

$$\Rightarrow \phi_{i-1}(d_1, d_2)$$

Path of len. $\leq 2^i$ between c, c'.

① $\phi_{i-1}(c, c'')$ ✓

② $\phi_{i-1}(c'', c')$ ✓

③ $F \Rightarrow \phi_{i-1}(d_1, d_2)$

$\boxed{= T}$

$$\phi_k(c_{start}, c_{accept})$$

$\|$

ϕ

$$|\phi_i| \le |\phi_{i-1}| + O(s(n))$$

$$\Rightarrow |\phi_{\underbrace{k}_{=O(s(n))}}| \le O(s(n)^2) \checkmark$$

$$|\phi_0| = O(s(n)) \quad \square$$

* we actually showed

TQBF is NPSPACE-complete

$\Rightarrow$ PSPACE = NPSPACE

Savitch's Theorem

Fact: $DSPACE(f) \subseteq NSPACE(f)$

Theorem: $\forall$ space constructible s,

$NSPACE(s) \subseteq DSPACE(s^2)$

Proof: Let $L \in NSPACE(s)$ w/ NTM M_L

- Let $G_{M_L,x}$ be the config graph of $M_L(x)$

Fact: # nodes in $G_{M_L,x}$

$\leq 2^{O(s(n))}$ for $x \in \{0,1\}^n$

Fact: $M_L(x) = 1 \iff$

$\exists$ path from c_{start} to

c_{accept} in $G_{M_L,x}$.

- Goal: Construct DTM

D_L which decides

L in space $O(s^2)$

$D_L(x)$:

- simulate $M_L(x)$ by

traversing

$G_{M_L,x}$

- Run recursive procedure

$\text{reach}(u, v, i)$

$= 1$ iff.

$\exists$ Path from u to v of length $\leq 2^i$

$\Updownarrow$

$\exists\ z$ s.t.

$\text{reach}(u, z, i-1) = 1$

and $\text{reach}(z, v, i-1) = 1$

- Run

output $\Rightarrow$ $\text{reach}(c_{start}, c_{accept}, \underbrace{O(s(n))}_{= k})$

↳ Enumerate over all

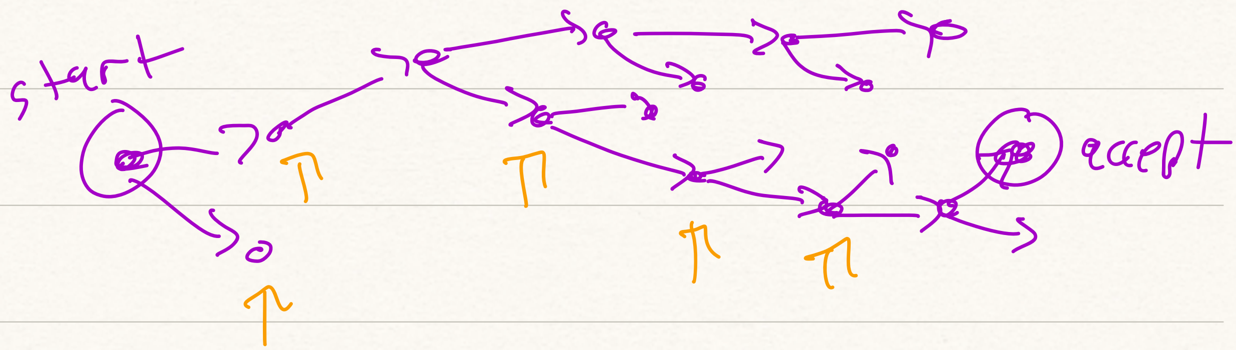
d (nodes) $\leftarrow 2^k$ nodes

↳ check if $\quad$ k space

reach(cstart, d, k-1) = 1

and

reach(d, caccept, k-1) = 1



→ recursion bottoms out after k steps

⇒ $O(k^2)$ space

⇒ $O(s(n)^2)$ space □

$$\underbrace{DSPACE(s)}_{n^c} \subseteq \overset{n^c}{NSPACE(s)} \subseteq \underbrace{DSPACE(s^2)}_{n^{2c}}$$

$\square$

<u>NL-Completeness</u>

$NL = \{ L \mid L$ is decidable on NTM in at most $O(\log(n))$ space $\}$

<u>Know</u> $PATH \in NL$

$\text{PATH} = \{(G, s, t) : \begin{array}{l} G \text{ is directed} \\ \text{graph w/} \\ \text{path } s \mapsto t \end{array}\}$

Unknown $\text{PATH} \in L$?

If $\text{PATH} \in L$ then
$L = NL$. ✗
→ we believe $L \neq NL$

PATH is NL-complete

* We need a new notion of reducibility

$\forall L, L' \in NL \setminus \{\emptyset, \{0,1\}^*\}$

$L \leq_p L'$

why?

- Decide on NTM in $O(\log(n))$ space
- Construct $2^{O(\log(n))}$ config. graph in poly-time.

Need: logspace-reducible.

Def: Let $f: \{0,1\}^* \to \{0,1\}^*$. Then f is (implicitly) logspace computable if $\exists\, c \geq 1$ s.t. $|f(x)| \leq |x|^c \;\forall x$ and the languages

$$L_f = \{(x,i) : f(x)_i = 1\}$$

$$L'_f = \{(x,i) : i \le |f(x)|\}$$

are in $\mathbf{L}$ (logspace).

- Def: Let B, C be languages. B is logspace reducible to C, denoted as $B \le_{\mathbf{L}} C$ if $\exists$ logspace f s.t.

$$x \in B \iff f(x) \in C.$$

- Def. B is NL-complete if

1. $B \in NL$
2. $\forall C \in NL$

   $C \le_L B$

<u>theorem:</u>

1. $A \le_L B$, $B \le_L C$, then

   $A \le_L C$.

2. If $A \le_L B$, $B \in \mathbf{L}$, then

   $A \in \mathbf{L}$

   $\hookrightarrow$ if $PATH \in \mathbf{L}$

   then $\mathbf{NL} = \mathbf{L}$

<u>Theorem</u> PATH is NL-complete.

<u>Proof</u>. $PATH \in NL$ shown before.

For NL-hardness

Show $B \leq_L PATH$.

Let $B \in NL$. Then $\exists$ NTM M_B s.t. $\forall x \in \{0,1\}^n$

$x \in B \iff M_B(x) = 1$ in $O(\log(n))$ space.

- From $M_B(x)$
  ↳ construct the config graph

$\{c_{accept}, c_{start}, G_{M_B, x}\} =: f(x)$

$|f(x)| \leq 2^{O(\log(n))}$
$\leq poly(n)$ ✓

- In $G_{M_B,x}$, $M_B(x)=1$
$\Longleftrightarrow$ $\exists$ path from $C_{start} \mapsto C_{accept}$.] PATH

- Represent $G_{M_B,x}$ by a $2^{O(\log(n))} \times 2^{O(\log(n))}$ adjacency matrix

- Entry $(c,c')=1$ iff there is an edge from c to c' in $G_{M_B,x}$

- This is logspace
  1. Given c, c'
     $\exists$ deterministic machine to check if

c' follows from c
according to M_B's
trasition function.

→ do this in space
$O(|c| + |c'|)$
$= O(\log(n))$ ✓ □