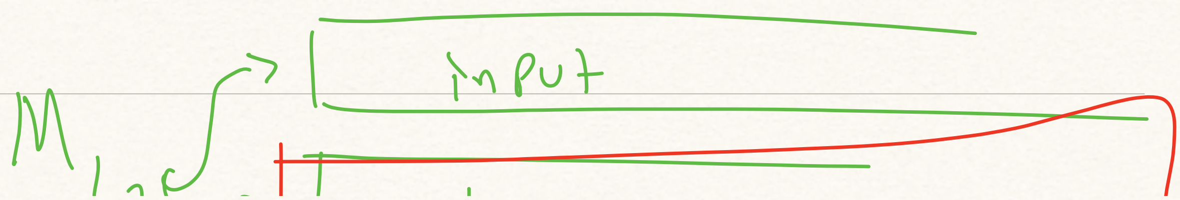
Space Complexity

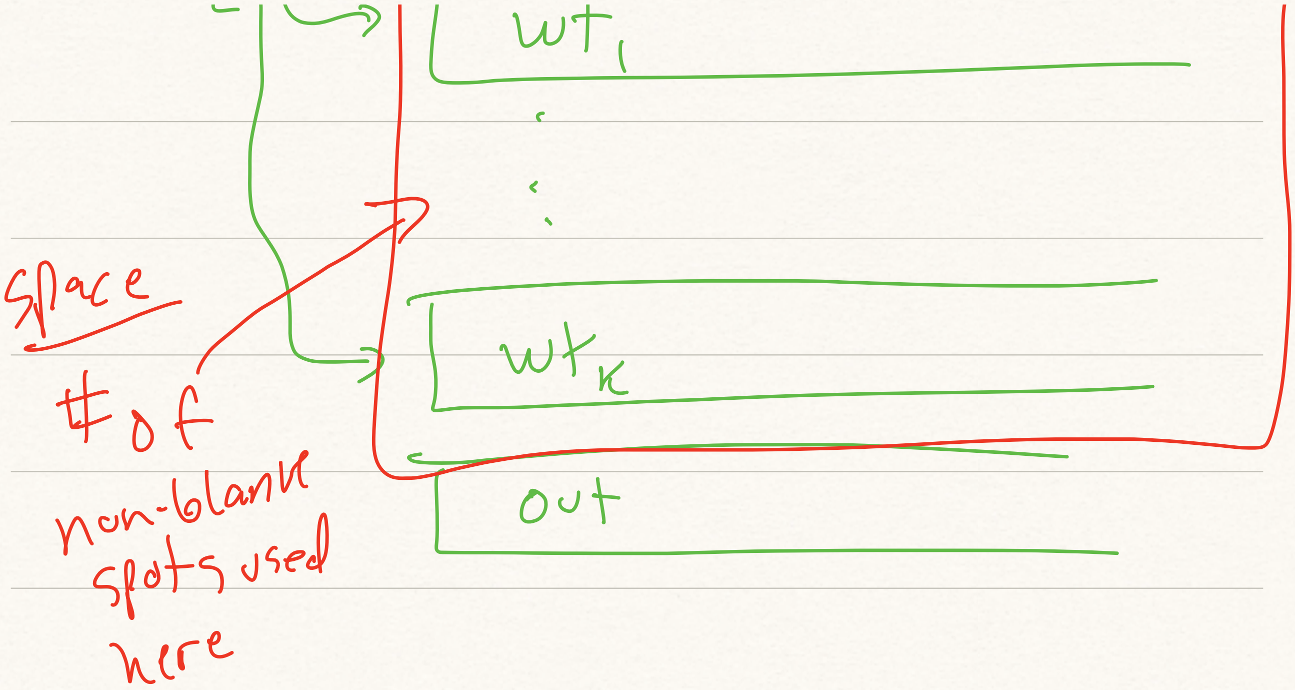
Lecture 9

<u>Def</u>. Let $S : \mathbb{N} \to \mathbb{N}$. Then a language $L \in DSPACE(S)$ if

$\exists$ a DTM M_L s.t.
for any $x \in \{0,1\}^*$
$M_L(x)$ decides L using <u>at most</u> $O(S(|x|))$ <u>additional</u> <u>space</u>

space on worktape





- Similarly, $L \in NSPACE(S)$ if $\exists$ NTM M_L w/ same space requirements
  - ↳ only uses $O(S(|x|))$ space for any computational path.

- **Ex:** $SAT \in DSPACE(n)$

Let ϕ be a formula
w/ m variables $x_1, \dots, x_m$
and $|\phi| = n$

- Let $\hat{M}$ be a DTM

$\hat{M}(\phi)$:

need $O(m)$ space
$m \leq n$

- $\boxed{\text{For all } u_1, \dots, u_m \in \{0,1\}}$
  - Test if $\underline{\phi(u_1, \dots, u_m)} = 1$
  - If yes, output 1
- Else output 0

$O(n)$ space is used

Space Constructible:

$S: \mathbb{N} \to \mathbb{N}$ s.t. $S(n) \geq \log(n)$

is space-constructible if

$\exists$ DTM M s.t. on input x

$M(x)$ outputs $S(|x|)$ in

at most $O(S(|x|))$

space.

Theorem: For every space

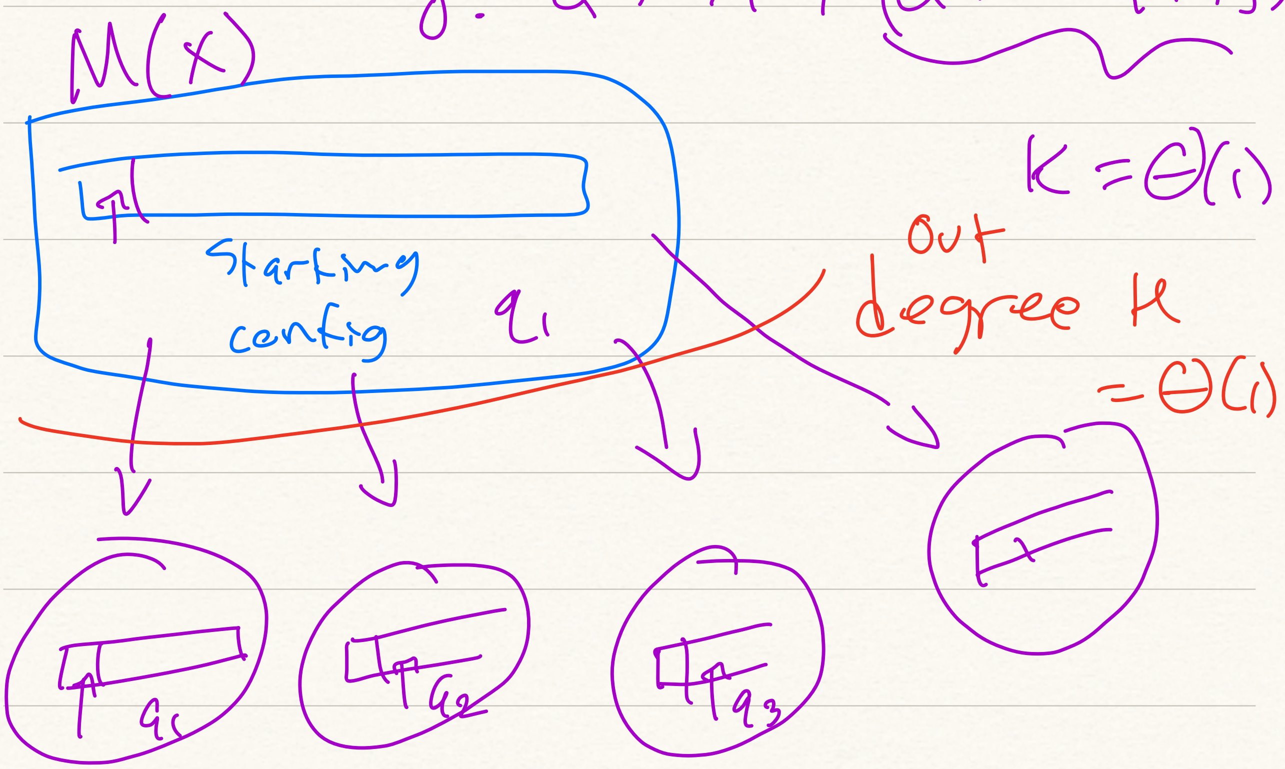
constructible S:

$$
\begin{aligned}
\mathrm{DTIME}(S) &\subseteq \mathrm{DSPACE}(S) \\
&\subseteq \mathrm{NSPACE}(S) \\
&\subseteq \mathrm{DTIME}(2^{O(S)})
\end{aligned}
$$

Show: $\mathrm{NSPACE}(S) \subseteq \mathrm{DTIME}(2^{O(S)})$

Turn NTM computation into a graph

$$\delta: Q \times \Gamma \to P(\underbrace{\overset{O(1)}{Q} \times \overset{\{0,1\}}{\Gamma} \times \overset{3}{\{L,R,S\}}}_{})$$

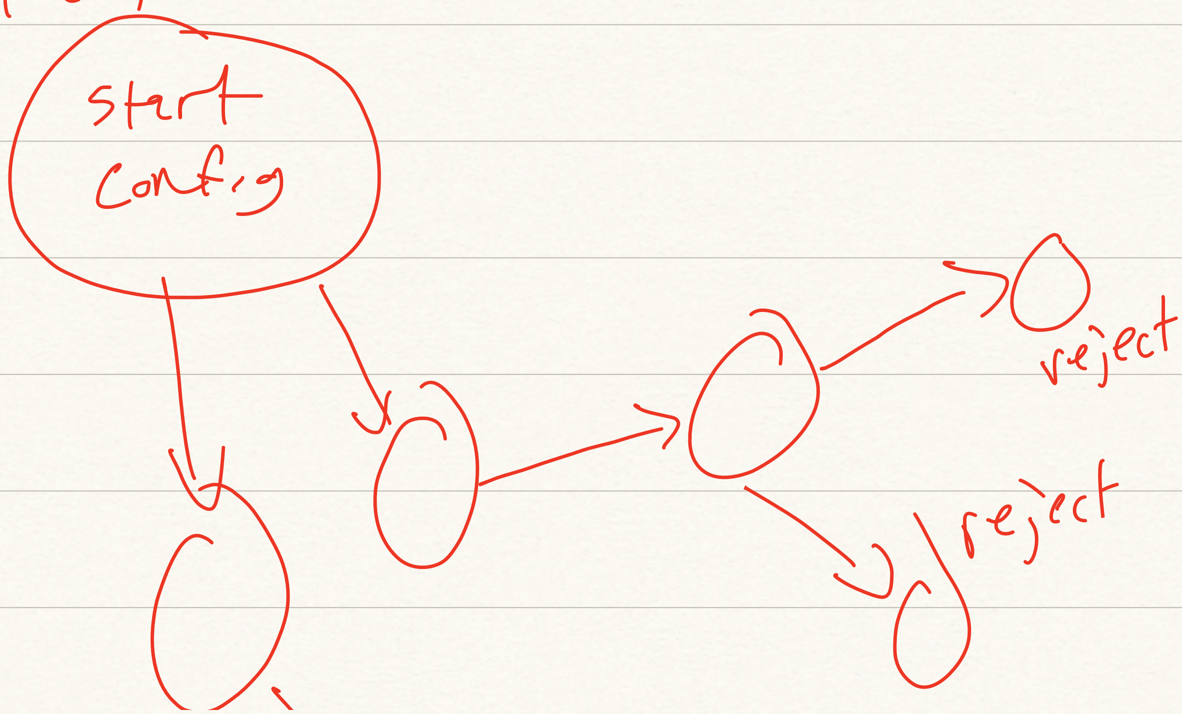


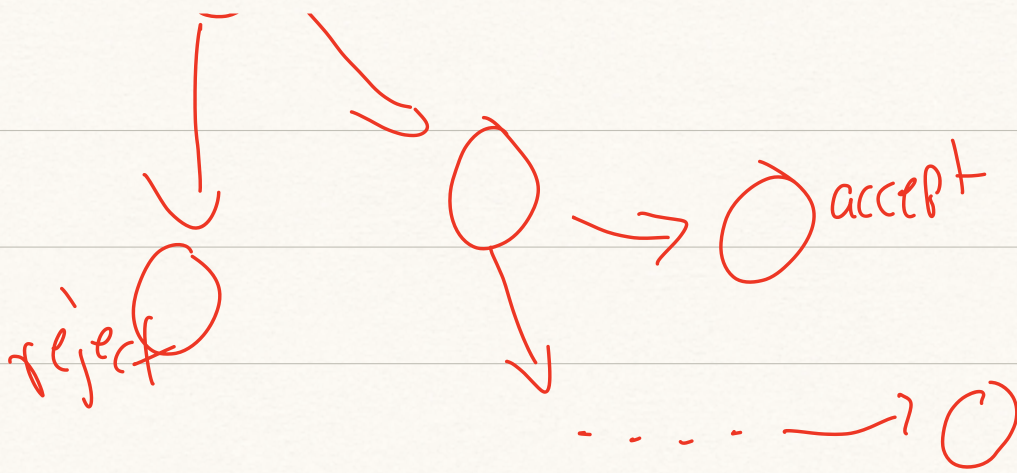
Known: You can convert any NTM w/ at most K outputs of its trasition

function to an equivalent
one with at most 2
outputs.

If $G_{M,x}$ (graph representing $M(x)$) has a path
from start to accept
then $M(x)$ accepts

$M(x)$





$G_{M,x}$: If M uses $O(s)$ space on any input, then

① $G_{M,x}$ has at most $2^{O(s)}$ nodes.

$\leftrightarrow$ only need $O(s)$ space to encode all configurations

② there exists an $O(s)$ sized CNF ψ such that

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

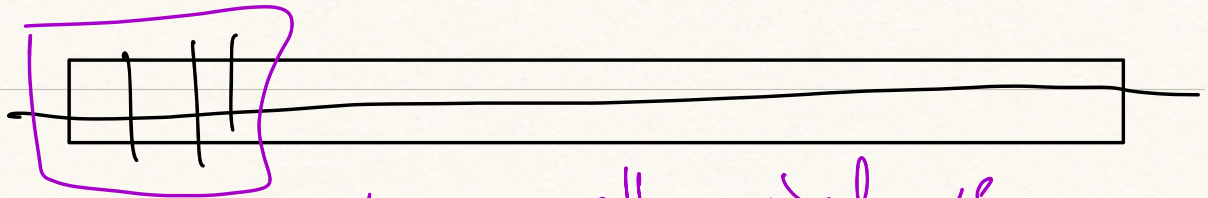
$c \mapsto c'$

Valid config

valid transition

valid config

$\Longrightarrow$ Cook-Levin



check over all windows
+ check symbols are valid

Back to the point

$$\underbrace{NSPACE(s)}_{\downarrow} \subseteq \underbrace{DTIME(2^{O(s)})}_{\downarrow}$$

TM M
that decides
some lang.
here

- $M'(x)$ gets
  M,
- Construct
  $G_{M,x}$
  $\hookrightarrow 2^{O(s)}$
- Test validity
  & find
  accept state
  w/ φ
  $\Rightarrow 2^{O(s)}$ time $\square$

<u>Space Hierarchy</u>

Let f, g be space constructible s.t. $f = o(g)$. Then

$DSPACE(f) \subsetneq DSPACE(g)$.

→ Proof identical to DTIME

EXCEPT: $f = o(g)$
rather than $f = o(g/\log g)$

→ no log overhead b/c
you can do universal
TM simulation w/ only
constant space overhead.

Space Complexity Classes

① $PSPACE = \bigcup_{c \in \mathbb{N}} DSPACE(n^c)$

$c \in \mathbb{N}$

↳ Space analogue of P

② $\text{NPSPACE} = \bigcup_{c \in \mathbb{N}} \text{NSPACE}(n^c)$

↳ space analogue of NP

③ $L = \text{DSPACE}(\log(n))$

④ $NL = \text{NSPACE}(\log(n))$

$L = NL$?. Open

$\text{PSPACE} \neq L$

$\text{NPSPACE} \neq NL$

$$\boxed{\text{PSPACE} = \text{NPSPACE}}$$

surprising

$\text{coNL} = NL$

Examples

$SAT \in PSPACE$

$\rightarrow$ $NP \subseteq PSPACE$

$L \in NP$

$\rightarrow$ TM M s.t. $\forall x$

$x \in L \longleftrightarrow \boxed{\exists w \in \{0,1\}^{p(|x|)} \quad M(x,w)=1}$

↳ iterate over all w in poly-space

Langs in L

$EVEN = \{ x \in \{0,1\}^* : x \text{ has even \# 1s} \}$

- $x \in \{0,1\}^n$

  $M(x)$:

  - count # 1s in binary

  counter max size in binary: $\lfloor \log(n) \rfloor + 1$

- $MULT = \{ (\langle x \rangle_2, \langle y \rangle_2, \langle z \rangle_2) :$
  $x, y, z \in \mathbb{N}$
  $xy = z \}$

- NL language

PATH $= \{ (G, s, t) :$ G is directed $\}$

graph w/ path from s to t

\# vertices/nodes = n

- If ∃ a path from s ↦ t, it is of length at most n
- Non-det go down paths of length n

BFS

PSPACE-Completeness

- A language L is

PSPACE complete if

1. $L \in$ PSPACE
2. $\forall L' \in$ PSPACE
   $L' \leq_p L$.

   still poly-time

TQBF

- For now we've only seen "fixed" formulas
  $\phi = (x_1 \vee x_2) \wedge (\bar{x}_3 \vee x_2)$

- $\phi \in SAT$
  $\hookrightarrow \exists\, x_1 x_2 x_3 x_4$ s.t.
  $\phi(x_1 x_2 x_3 x_4) = 1$

$\phi(x_1 \wedge x_2 \wedge x_3 \wedge x_4) = 1$

$$\exists x_1 x_2 x_3 \quad \phi(\vec{x})$$

$$\exists x_1 x_2 \forall x_3 \ \phi(x_1 x_2 x_3) = 1$$

$$(x_1 \vee x_2) \wedge (\bar{x}_3 \vee x_2)$$

0 1 0/1 1

Let ϕ be an m-variable Boolean Formula. Then

$$\underline{Q_1 x_1 Q_2 x_2 \cdots Q_n x_n} \ \phi(x_1, \ldots, x_m)$$

is a <u>quantified</u> Boolean formula.

$Q_i \in \{\exists, \forall\}$ for all i.

$x_i \in \{0,1\}$

$Q_1 x_1 (\psi \wedge \phi)$

Examples

$\exists x_1 \exists x_2 \ldots \exists x_m$

$\exists x \; \phi(x) = 1 \Rightarrow SAT$

$\forall x \; \phi(x) = 1 \Rightarrow TUAT$

$\forall x \; \phi(x) = 0 \Rightarrow UNSAT$

$\forall x \exists y \; (x \wedge y) \vee (\bar{x} \wedge \bar{y})$

$(x == y) = TRUE$

$\exists x \forall y \; \boxed{(x == y)} \rightarrow FALSE$

$TQBF = \{ \psi : \psi \text{ is a true QBF} \}$

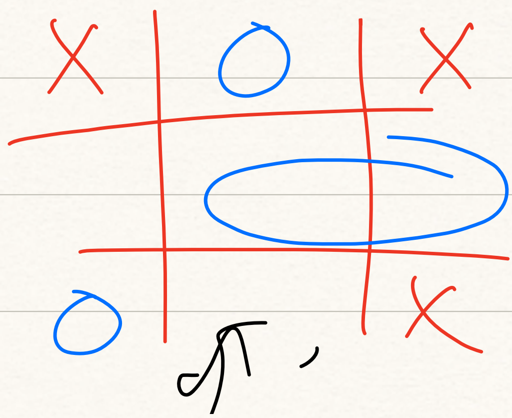
Thm. TQBF is PSPACE complete.

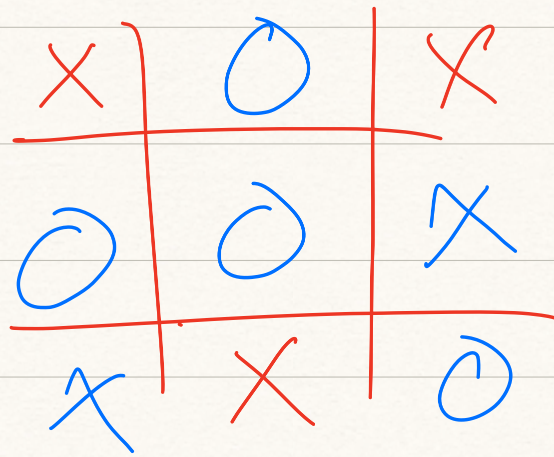
Aside **Essence of PSPACE**

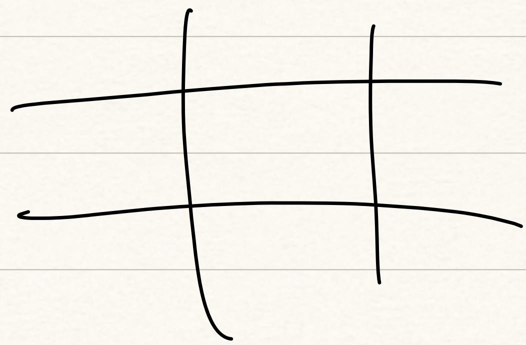
↓

Finding optimal solutions for 2 player games.

Ex: Chess or Go







① TQBF $\in$ PSPACE

- Let ψ be a QBF

  $\underset{x_1, \ldots, x_m}{\psi}$ $\quad |\psi| = n$

$M(\psi)$:

① If ψ has no quantifiers then check if $\psi = 1$ or 0. Output the same.

② Else if $\psi = \underline{\exists} x \underbrace{\phi}_{\text{QBF}}$

- Run $M(\phi)$ with
  $x = 0$ and $x = 1$.
- If either return 1, return 1. Else return 0

③ Else if $\psi = \forall x\, \phi$
- Run $M(\phi)$ with $x = 0$ and $x = 1$
- Return 1 iff both return 1.

Recursion Depth: at most $m \le n$

Every level: storing $\le m$ information

$\Rightarrow$ ~~$O(n^2)$ space.~~

$\Rightarrow$ $O(m)$ space

Lecture 9