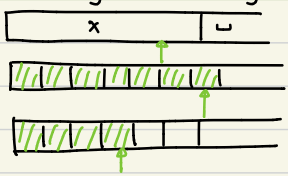
5.7 Space Complexity

resource: time, space, randomness, parallelism, etc.

Def 5.30 Let $S: \mathbb{N} \to \mathbb{N}$. A TM M is said to run in space $S(n)$ if, for every input x, it uses at most $S(|x|)$ tape cells on its work tape, excluding read-only input tape.

input tape:

work tapes:



Def 5.31 Let $S: \mathbb{N} \to \mathbb{N}$. Language $L \in SPACE(S(n))$ if there is a TM M that runs in space $O(S(n))$ and decides L.

$PSPACE \overset{def}{=} SPACE(n^{O(1)})$

$L \overset{def}{=} SPACE(\log n)$

$SAT = \{\langle\phi\rangle : \phi \text{ has a satisfying assignment}\}$.

$SAT \in$ NP-complete

$SAT \in SPACE(n)$

time: 2^n $\quad \phi = (x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_2 \vee x_5 \vee \bar{x}_7) \wedge \cdots$

For every $\varphi: \{x_1, \cdots, x_n\} \to \{0,1\}$, check if ρ is a satisfying assignment

Def 5.32 Let $S: \mathbb{N} \to \mathbb{N}$. Language L is in $NSPACE(S(n))$ if there is an NTM that runs in space $O(S(n))$ and decides L.

$NL \overset{def}{=} NSPACE(\log n)$

UTM 模拟：时间 × log
空间未变大.

Def 5.33 (Space constructible function) Function $S: \mathbb{N} \to \mathbb{N}$ is space constructible if there is a TM that, on input 1^n, outputs the binary representation of $S(n)$ in space $O(S(n))$.

eg. $\lfloor \log_2 n \rfloor$, n, n^2, 2^n, are space constructible.

Thm 5.34 If f and g are space-constructible functions satisfying $f(n) = o(g(n))$, $(f(n) \ll g(n))$ then $SPACE(f(n)) \subsetneq SPACE(g(n))$

Proof. Construct the following TM: on input $x \in \{0,1\}^*$, simulate M_x on input x up to $S(n)$, where $S(n) \overset{def}{=} f(n) \times (\to \infty)$

$$= f(n) \cdot \lfloor \log_2 \frac{g(n)}{f(n)} \rfloor$$

$s(n)$ is space constructible in space $O(g(n))$.

$s(n) = o(g(n))$. $S(n) = \omega(g(n))$.

For each work tape, mark the $s(n)^{th}$ cell using a special symbol @.

Once @ is read, terminate M_x.

$\#\text{configurations} \le |Q| \times (|\Gamma|^{s(n)})^m \times S(n)^m$. m tapes.

Count the number of steps, if it exceeds the above number, stop and accept. If the special symbol is read, stop and accept.

Otherwise, flip the output of M_x.

We claim $L \in SPACE(g(n))$, and $L \notin SPACE(f(n))$. □

Let $S: \mathbb{N} \to \mathbb{N}$ where $S(n) \ge \log_2 n$.

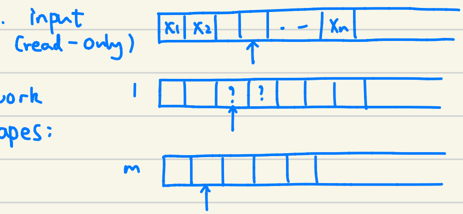
Thm 5.35 1) $DTIME(S(n)) \subseteq SPACE(S(n))$.

2) $SPACE(S(n)) \subseteq NSPACE(S(n))$. // NTM 比 TM 更强大

3) $NSPACE(S(n)) \subseteq DTIME(2^{O(S(n))})$. // 画出状态图.

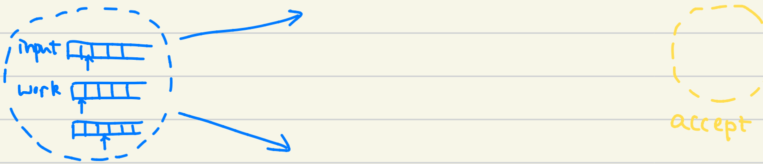
$O_M(S(n))$.

Proof.



#bits to encode a config is $O(\log|Q| + \log n \times (m+1) + S(n) \times m \times \log|\Gamma|)$
$= O_M(\log n + S(n))$
$= O_M(S(n))$.

Let $G_{M,x}$ be the configuration graph for NTM M with input $x \in \{0,1\}^*$



Use BFS to check if s and t are connected in time
$O(|V|+|E|) = O_M(|V|) = O_M(2^{\#\text{bits}}) = O_M(2^{O_M(S(n))}) = 2^{O_M(S(n))}$ $\square$

Thm 5.36 (Savitch's Theorem) Let $S: \mathbb{N} \to \mathbb{N}$, where $S(n) \geq \log_2 n$.
Then $\mathrm{NSPACE}(S(n)) \subseteq \mathrm{SPACE}(S(n)^2)$.

The theorem says deterministic TM can simulate nondeterministic TM by using a quadratic amount of original space.

Proof. Let $L \in \mathrm{NSPACE}(S(n))$. prove $L \in \mathrm{SPACE}(S(n)^2)$

By definition, L is decided by an NTM M that runs in space $O(S(n))$

consider the configuration graph $G_{M,x}$ where $x \in \{0,1\}^*$ is the input.

1) Each vertex $v \in V(G_{M,x})$ is described by $O_M(S(n))$ bits.

2) $|V(G_{M,x})| = 2^{O_M(S(n))}$.

3) Each vertex has ≤ 2 outgoing edges. Since we can assume M is a binary-choice NTM.

4) Two outgoing neighbours can be computed in time $O_M(S(n))$.

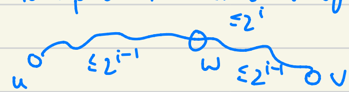
5) No self loop.

6) $G_{M,x}$ has one start state s.

7) $G_{M,x}$ has one accept state t. Do the cleaning after M accepts x

M accepts x iff. there is a path from s to t.

Define $\text{REACH}(u,v,i)=1$ if there is a path from u to v of length $\leq 2^i$.



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

If $i=0$, decide if $(u,v)\in E(G_{M,x})$ and return

for $w\in V(G_{M,x})$

if $\text{REACH}(u,w,i-1)$ and $\text{REACH}(w,v,i-1)$

then true

return false

recursion depth: $\log|V| = O_M(S(n))$

space per node $O_M(S(n))$.

total $O_M(S(n)^2)$

$\square$

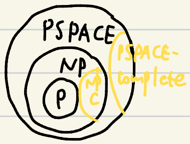
directed s-t connectivity

	BFS	DFS
time	$O(\lvert V\rvert+\lvert E\rvert)$	$O(\lvert V\rvert+\lvert E\rvert)$
space	$O(\lvert V\rvert)$	$O(\lvert V\rvert)$

$|V| = 2^{O_M(S(n))}$.

time: $(\text{fan-out})^{\#\text{depth}}$

$=|V|^{\log|V|}$



5.10 PSPACE problem

Recall that $PSPACE = SPACE(n^{O(1)})$

Def 5.37 A language L is PSPACE-complete if

1. $L \in PSPACE$, and
2. $(\forall K \in PSPACE)(K \leq_p L)$.

游戏

PSPACE-Complete problems

证明必胜策略

1) Generalized Tic-Tac-Toe
2) Generalized Go
3) Generalized Geography
4) Planning with constraints
5) Equivalence of regular expressions with complement

Def 5.38 (TQBF) A quantified Boolean formula (QBF) is an expression of the form $Q_1 x_1 Q_2 x_2 \cdots Q_n x_n \varphi(x_1, \cdots, x_n)$, where each $Q_i \in \{\forall, \exists\}$, each x_i is a Boolean variable, and φ is Boolean formula.

$\exists x_1 \exists x_2 \cdots \varphi(x_1 \cdots)$ SAT NP

$\forall x_1 \forall x_2 \cdots \varphi(\cdot\cdot)$ coNP

$TQBF = \{\langle \phi \rangle : \phi \text{ is a true QBF}\}$

$((x_1 \vee \bar{x}_2) \wedge x_3) \vee \bar{x}_5$

A Boolean formula is an expression built from Boolean variables, the constants 0 and 1, the connectives $\vee, \wedge, \neg$

$\forall x \varphi(x)$ means $\varphi(x)$ holds for every $x \in \{0,1\}$.

$\exists x \varphi(x)$ means $\varphi(x)$ holds for at least one $x \in \{0,1\}$.

Example. $\varphi(x,y) = (x \vee \neg y) \wedge (\neg x \vee y)$

$= \neg(x \oplus y)$

x	y	$(x \vee \neg y) \wedge (\neg x \vee y) = \overline{x \oplus y}$
0	0	1
0	1	0
1	0	0
1	1	1

$\forall y \, \exists x \, \varphi(x, y)$ true $\in TQBF$

$\exists x \, \forall y \, \varphi(x, y)$ false $\notin TQBF$

∃ 玩家：想为 true　∀ 玩家：想为假. 必胜策略?

把 Tic-Tac-Toe 归约到 ← TQBF

QBFEVAL benchmark competition

Example. Tic-Tac-Toe

	O_2	O_4
	x_1	
x_3		

0000	0001	0010
0011	0100	0101
0110	0111	1000

$9 \le 2^4$ $(a_i, b_i, c_i, d_i) \in \{0,1\}^4$

$(\exists a_1 \exists b_1 \exists c_1 \exists d_1)(\forall a_2 \forall b_2 \forall c_2 \forall d_2) \cdots (\exists a_9 \exists b_9 \exists c_9 \exists d_9)(\varphi(a_1, \cdots, d_9))$

Formula $\varphi(a_1, b_1, \cdots, a_9, b_9, c_9, d_9)$ enforces the rules of the game:

1) Validity. each move encodes a valid board position

2) Uniqueness: no two moves select the same position

3) Winning condition

Theorem 5.39 TQBF is PSPACE-complete.

Proof. First, we prove TQBF $\in$ PSPACE.

Let $\psi = Q_1 x_1 Q_2 x_2 \cdots Q_n x_n \underline{\phi(x_1, \cdots, x_n)}$ → no other free variables

be a QBF with n variables and of size m.

$\text{TQBF}(\psi = Q_1 x_1 \cdots Q_n x_n \, \varphi(x_1 \cdots x_n))$

if $n=0$, evaluate φ and return φ.

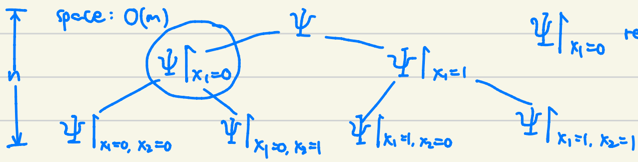
let $Q_1 x_1$ be the first quantifier

Define $\psi_0 = Q_2 x_2 \cdots Q_n x_n \, \varphi(0, x_2, \cdots, x_n)$

$\psi_1 = Q_2 x_2 \cdots Q_n x_n \, \varphi(1, x_2, \cdots, x_n)$

If $Q_1 = \exists$, return $\text{TQBF}(\psi_0) \vee \text{TQBF}(\psi_1)$

If $Q_1 = \forall$, return $\text{TQBF}(\psi_0) \wedge \text{TQBF}(\psi_1)$



$\Psi|_{x_1=0}$ restricting x_1 to 0 in Ψ.

space: $O(nm)$ time: $O(2^n)$

Space complexity: (recursion depth) $\times$ (space per node)

$= n \times O(m) = O(mn) = O(|x|^2)$

input size $= O(m+n)$

So, TQBF $\in$ PSPACE.

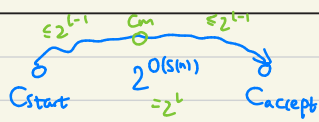
We show $L \leq_p$ TQBF for every $L \in$ PSPACE.

By definition, L is decided by a TM that runs in space $S(n)$.

where $S(n)$ is a polynomial

$x \overset{?}{\in} L \xrightarrow{\text{polytime}} \Psi = \Psi(x, M) \quad \overset{?}{\in} \text{TQBF}$

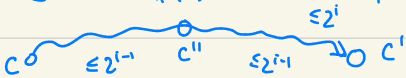
Let $x \in \{0,1\}^*$. We construct QBF $\Psi = \Psi(M, x)$ of size $O_M(S(n)^2)$, such that $x \in L$ iff. $\Psi \in$ TQBF.



Let c denote a configuration, where $C \in \{0,1\}^{O_M(S(n))}$

Let $\Psi_i(C, C')$ is true if there is a path from c to c' of length $\leq 2^i$

Finally, $\Psi = \Psi_{O_M(S(n))}(C_{start}, C_{accept})$



$\Psi_i(C, C') = (\exists C'')(\Psi_{i-1}(C, C'') \wedge \Psi_{i-1}(C'', C'))$ 大小不对. $S_i = 2S_{i-1}$

$\Psi_i(C, C') = (\exists C'')(\forall D)(\forall E)$

$((D = C \wedge E = C'') \vee (D = C'' \wedge E = C') \rightarrow \Psi_{i-1}(D, E))$

$S_i = S_{i-1} + O_M(1) \Rightarrow S_{O_M(S(n))} = O_M(S(n)) + O_M(S(n)^2) = O_M(S(n)^2)$

$\Psi_0(C, C') = \text{true}$ if C' is the next configuration of C.

We have done this in Cook-Levin theorem, the size of the formula is $O_M(S(n)^2)$.

$\square$