PSPACE

- Savitch's algorithm $O(\log^2 n)$ space.
  $NL \subseteq L^2$
- $NP \subseteq PSPACE$
- $L \subseteq P$, $PSPACE \subseteq EXP$
- $NL = coNL$ ($NP \stackrel{?}{=} coNP$)
- $NPSPACE = coNPSPACE$ $\quad$ $NPSPACE = PSPACE$
  $NSPACE(s(n)) = coNSPACE(s(n))$.

TQBF : Totally quantified boolean formula

$$\psi = \exists x_1 \forall x_2 \exists x_3 \forall x_4 \cdots \exists x_n \underbrace{\phi(x_1, \ldots, x_n)}_{3\text{-CNF}}$$

$$\psi = \exists x_1 x_2, \ldots, x_n \ \phi(x_1, \ldots, x_n)$$

$$\Leftrightarrow \exists x_1 \forall y_1 \exists x_2 \forall y_2 \cdots \exists x_n \ \phi(x_1, \ldots, x_n)$$

$TQBF(\psi)$ = value of ψ.

$$3\text{-SAT}(\phi) = TQBF(\psi) \qquad \psi = \exists x_1 \exists x_2 \ldots \exists x_n \ \phi(x_1, \ldots, x_n)$$

Lemma: $TQBF \in PSPACE$

Pf:

$$\psi = \exists x_1 \forall x_2 \ldots \exists x_n \ \phi(x_1, \ldots, x_n)$$

$$\psi_1 = \forall x_2 \ldots \exists x_n \ \phi(1, x_2, \ldots, x_n)$$

$$\psi_2 = \forall x_2 \ldots \exists x_n \ \phi(0, x_2, \ldots, x_n)$$

$$\psi = \psi_1 \vee \psi_2$$

$$\psi = \forall x_1 \exists x_2 \cdots \phi(\ \ )$$

$$\psi_1 = \ldots \ \phi(1, \ldots)$$

$$\psi_2 = \ldots \ \phi(0, \ldots)$$

$$\psi = \psi_1 \wedge \psi_2$$

Alg

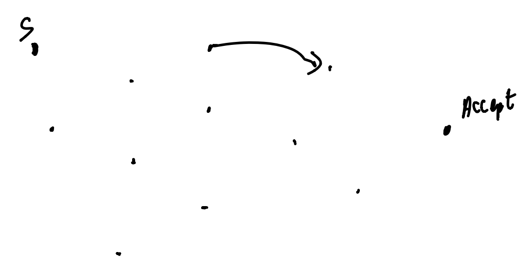
- Recursively compute ψ_1 → Use same space
- " " ψ_2 → Use same space
- Output $\psi_1 \vee \psi_2$

Claim: Space complexity $O(m \cdot n)$ where n : # variables, m : size of formula.

$$O(m) + O((n-1) \cdot m) \le O(n \cdot m).$$

Thm: For every $f \in PSPACE$, there is a poly time computable g such that $f(x) = TQBF(g(x))$.

Pf: $G_{M,x}$



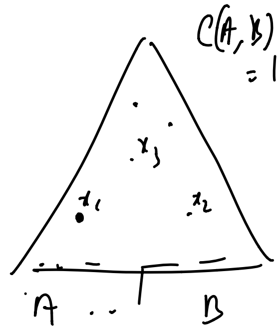
$$2^{poly(n)} = 2^{O(s(n))}$$

Goal: Convert the code of machine M + x → $\underbrace{\text{TQBF formula}}_{poly(n)}$.

$$\psi_i(A,B) = \begin{cases} 1 & \text{if there is a path of length} \le 2^i \text{ from } A \text{ to } B \\ 0 & \text{o.w.} \end{cases}$$

$\psi_0(A,B)$

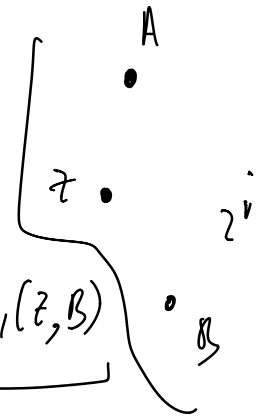
1. Make a circuit $C(A,B) = \begin{cases} 1 & \text{if } M(x) \text{ goes from } A \to B \text{ in a single step} \\ 0 & \text{o.w.} \end{cases}$
2. Make a $\phi(x_1, \dots, x_n)$ s.t. $\exists x_1, \dots, x_n \; \phi(x_1, \dots, x_n) \iff C(A,B)$.



$\iff \quad \exists x_1,\dots,x_n \ \phi(x_1,\dots,x_n)$

WANT

$\psi_i(A,B)$ using $\psi_{i-1}(X,Y)$



X Doubles the size!

$$\psi_i(A,B) = \underbrace{\exists Z \ \psi_{i-1}(A,Z) \wedge \psi_{i-1}(Z,B)}$$

$$\psi_{i-1}(A,Z) = \exists x_1 \forall x_2 \cdots \exists x_n \ \phi(x_1,\dots,x_n)$$

$$\psi_{i-1}(Z,B) = \exists y_1 \forall y_2 \cdots \exists y_n \ \phi'(y_1,\dots,y_n)$$

$$\psi_i(A,B) = \exists z_1,\dots,z_m \ \exists x_1 \forall x_2 \dots \exists x_n \exists y_1 \forall y_2 \dots \exists y_n$$
$$\phi(x_1,\dots,x_n) \wedge \phi'(y)$$

$$\psi_i(A,B) = \exists Z \forall X \forall Y \ (X=A \wedge Y=Z) \vee (X=Z \wedge Y=B)$$
$$\Rightarrow \psi_{i-1}(X,Y)$$

$$= \exists Z \forall X \forall Y \left(\neg(X=A \wedge Y=Z) \wedge \neg(X=Z \wedge Y=B)\right)$$
$$\vee \ \psi_{i-1}(X,Y)$$

$$= \exists Z \, \forall X \, \forall Y \, \exists W \; \phi(Z, X, Y, W)$$

Size $O(i \cdot \text{poly}(n))$.

$\blacksquare$

<u>Theorem</u>: There is no algorithm for 3-SAT running in $O(n)$ time and $O(\log n)$ space.

<u>Pf</u>: Show that such an algorithm

$$\Rightarrow \quad DTIME(n^2) \subseteq DTIME(n \cdot \text{poly}\log(n))$$

$$\text{``} \quad n \cdot \log^{10}(n).$$

<u>Thm</u>: $f \in NTIME(t(n))$ can be reduced in log space and time $O(t(n) \log(t(n)))$ to computing 3-SAT on a formula of size $O(t(n) \log(t(n)))$.

$f \in DTIME(n^2)$

$f(x) = M_f(x)$.

$\underline{M}$

1. Generate formula ϕ of size $n^2 \log n$ simulating M_f.

2. Run 3-SAT algorithm on formula

M: runs in time $O(n^2 \log n)$

space $O(\log n)$

<u>Consider config graph of $G_{M,x}$</u>

$M(x) = \exists\, C_1, C_2, \ldots C_{\sqrt{n^2 \log n}}$

$\forall i$ C_i follows from C_{i-1} in $\sqrt{n^2 \log n}$ steps.