Definition 1

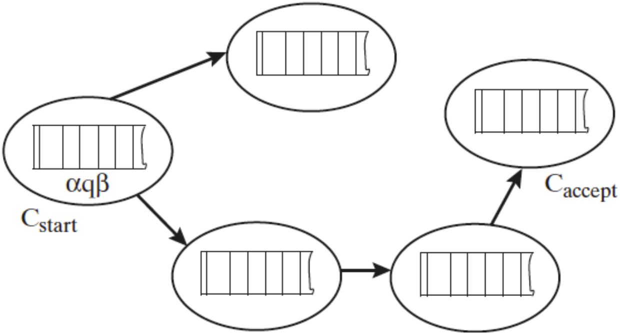
$L \in SPACE(s(n))$ if TM M decides L using space $O(s(n))$ in work tape(s). $L \in NSPACE(s(n))$ if NDTM M decides L using space $O(s(n))$ in work tape(s).

Note: $s(n)$ is space-constructible, i.e., can be computed in $O(s(n))$ space (a TM space-bounded by $s(n)$ can calculate how much space it uses).

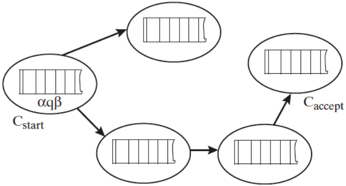
- $DTIME(t(n))$ doesn't make sense for $t(n) < n$
- $DSPACE(s(n))$ does make sense for $s(n) < n$ (e.g., $s(n) = \log n$)

Configuration graph $G_{M,x}$

- Nodes=Configurations. Snapshot of TM M with (work tape(s) contents, head position(s), state). If M space-bounded by $s(n)$, then at most $2^{O(s(n))}$(!!!) configurations.
- Edges (C, C') if M can go from configuration C to configuration C' in one step when input is x.



Configuration graph $G_{M,x}$



- Can make M to clean tapes & move heads to fixed positions before accepting, so only one accepting final configuration C_{accept}. C_{start} is starting configuration.
- If M DTM, then out-degree of each configuration is ≤ 1. If M NDTM, then out-degree of each configuration is ≤ 2 (two possibilities for the current bit of certificate).
- $M(x) = 1$ iff there is directed path $C_{start} \rightsquigarrow C_{accept}$
- $O(s(n))$-size CNF $\phi_{M,x}(C, C') = 1 \Leftrightarrow (C, C') \in E_{G_{M,x}}$ (Cook's theorem)

Theorem 2

$DTIME(s(n)) \subseteq SPACE(s(n)) \subseteq NSPACE(s(n)) \subseteq DTIME(2^{O(s(n))})$

Proof:

- $NSPACE(s(n)) \subseteq DTIME(2^{O(s(n))})$: Construct $G_{M,x}$ in $2^{O(s(n))}$ time, run BFS to see if there is path $C_{start} \rightsquigarrow C_{accept}$ in $2^{O(s(n))}$ time.
- $DTIME(s(n)) \subseteq SPACE(s(n)) \subseteq NSPACE(s(n))$ Obvious

□

Definition 3

$$PSPACE = \bigcup_{c \geq 0} SPACE(n^c)$$

$$NPSPACE = \bigcup_{c \geq 0} NSPACE(n^c)$$

$$L = SPACE(\log n)$$

$$NL = NSPACE(\log n)$$

Theorem 4

$NP \subseteq PSPACE$

Proof:

Run $L \leq_P 3SAT$ in poly-time (and poly-space) to compute CNF $|\phi_L(x)| = O(|x|^c)$ and try all possible assignments (of size $O(|x|^c)$). □

$PATH = \{\langle G, s, t\rangle :$ directed G has directed path $s \rightsquigarrow t\}$

Theorem 5

$PATH \in NL$

Proof:
NDTM that takes as certificate at most n nodes of $s - t$ path. We need indices for certificate and input, to check if $(cert_i, cert_{i+1}) \in E$. Each index of size $O(\log n)$. □

Note: Is $PATH \in L$? Open like $L \stackrel{?}{=} NL$ (naturally, because $PATH$ is NL-complete).
...But if G undirected then $PATH \in L$!

Theorem 6 (Space Hierarchy)

If f, g space-constructible with $\lim \frac{f(n)}{g(n)} = 0$ (i.e., $f(n) = o(g(n))$), then

$$SPACE(f(n)) \subset SPACE(g(n))$$

Proof:

Same as Time Hierarchy, but now Universal TM simulates SPACE(f(n)) in $O(f(n))$ (not $O(f(n) \log f(n))$ as in time hierarchy). □

Definition 7 (*PSPACE*-hardness)

L is *PSPACE*-hard if $L' \leq_P L$ for every $L' \in PSPACE$.

Definition 8 (*PSPACE*-completeness)

L is *PSPACE*-complete if

1. L is *PSPACE*-hard, and
2. $L \in PSPACE$.

$SPACE\ TMSAT = \{\langle M, x, 1^n \rangle : \text{DTM } M \text{ accepts } x \text{ in space } n\}$

Theorem 9

SPACE TMSAT is PSPACE-complete.

Definition 10 (Quantified Boolean Formula (QBF))

$$Q_1 x_1 Q_2 x_2 \dots Q_n x_n \; \phi(x_1, x_2, \dots, x_n)$$

where $Q_i = \exists$ or $\forall$, and ϕ is unquantified formula.

Note: Wlog we can assume that ϕ is 3CNF

QSAT. Let $\Phi(x_1, \ldots, x_n)$ be a Boolean CNF formula. Is the following propositional formula true?

$$\exists x_1 \ \forall x_2 \ \exists x_3 \ \forall x_4 \ldots \ \forall x_{n-1} \ \exists x_n \ \Phi(x_1, \ldots, x_n)$$

assume n is odd

Intuition. Amy picks truth value for x_1, then Bob for x_2, then Amy for x_3, and so on. Can Amy satisfy Φ no natter what Bob does?

Ex. $(x_1 \vee x_2) \wedge (x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3)$
Yes. Amy sets x_1 true; Bob sets x_2; Amy sets x_3 to be same as x_2.

Ex. $(x_1 \vee x_2) \wedge (\bar{x}_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3)$
No. If Amy sets x_1 false; Bob sets x_2 false; Amy loses;
if Amy sets x_1 true; Bob sets x_2 true; Amy loses.

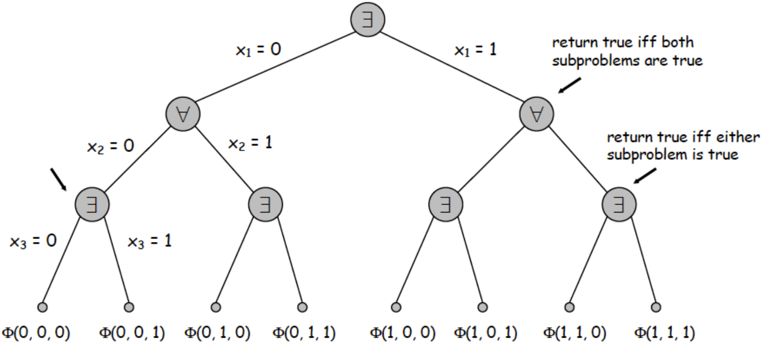
Algorithm $QBF(Q_1x_1Q_2x_2 \ldots Q_nx_n\ \phi(x_1, x_2, \ldots, x_n))$

if $Q_1 = \exists$ **then**
  return $QBF(Q_2x_2 \ldots Q_nx_n\phi(0, x_2, \ldots, x_n)) \vee QBF(Q_2x_2 \ldots Q_nx_n\phi(1, x_2, \ldots, x_n))$
else if $Q_1 = \forall$ **then**
  return $QBF(Q_2x_2 \ldots Q_nx_n\phi(0, x_2, \ldots, x_n)) \wedge QBF(Q_2x_2 \ldots Q_nx_n\phi(1, x_2, \ldots, x_n))$
else
  return 1

Theorem. QSAT ∈ PSAPCE.

Pf. Recursively try all possibilities.

- Only need one bit of information from each subproblem.
- Amount of space is proportional to depth of function call stack.



Theorem 11

TQBF (or QSAT) is PSPACE-complete.

Proof:

- $TQBF \in PSPACE$: Already shown
- $\forall L \in PSPACE : L \leq_P TQBF$: Write acceptance $M(x) = 1$ for L as $PATH$ problem in configuration graph $G_{M,x}$
  $\Rightarrow$ reduce $PATH(C, C')$ in 2^i steps to (recursive) QBF

$$\psi_i(C, C') = \exists C'' \; \psi_{i-1}(C, C'') \wedge \psi_{i-1}(C'', C)$$

...**but** ψ_m needs 2^m space (recurrence $S(i) = 2S(i-1) + O(1)$)
$\Rightarrow$ increase # of vars to get recurrence $S(i) = S(i-1) + O(1)$:

$$\psi_i(C, C') =$$
$$\exists C'' \forall D_1 \forall D_2 ((D_1 = C \wedge D_2 = C'') \vee (D_1 = C'' \wedge D_2 = C')) \Rightarrow \psi_{i-1}(D_1, D_2)$$

□

Configuration graph argument exactly the same for NDTM M for $L \in NPSPACE$!

Theorem 12

TQBF (or QSAT) is NPSPACE-complete.

Theorem 13

$PSPACE = NPSPACE$.

(compare with P vs. NP)

Theorem 14

$PSPACE = coPSPACE$.

$\Rightarrow \overline{TQBF}$ is also $PSPACE$-complete!

Theorem 15 (Savitch's theorem)

For $S(n) \geq \log n$, $NSPACE(S(n)) \subseteq SPACE(S(n)^2)$.

Proof:

1. Configuration graph $G_{M,x}$ for $L \in NSPACE(S(n))$ ($2^{O(S(n))}$ nodes)
2. $REACH?(u, v, i) = 1$ iff $\exists$ path $u \rightsquigarrow v$ of length $\leq 2^i$

Algorithm $REACH?(u, v, i)$

```
if u = v then
    return 1
for each node z do
    if REACH?(u, z, i − 1) = 1 ∧ REACH?(z, v, i − 1) = 1 then
        return 1
return 0
```

3. If $REACH?(C_{start}, C_{accept}, S(n)) = 1$ then $x \in L$.
4. Space recurrence:
$s(2^i) = s(2^{i-1}) + O(\log 2^{S(n)}) \Rightarrow s(S(n)) = O(S(n)^2)$ □

Definition 16 (implicitly logspace computable function)

Function f is implicitly logspace computable if $|f(x)| \leq n^c$ and $L_f = \{\langle x, i \rangle : f(x)_i = 1\}$, $L_f = \{\langle x, i \rangle : i \leq |f(x)|\}$ are in L.

Definition 17 (logspace reducibility $\leq_l$)

B is logspace reducible to C ($B \leq_l C$) if there is implicitly logspace computable function f such that $\forall x \in \{0,1\}^* : x \in B \Leftrightarrow f(x) \in C$.

Definition 18 (NL-complete)

C is NP-complete if $C \in NL$ and $\forall B \in NL : B \leq_l C$.

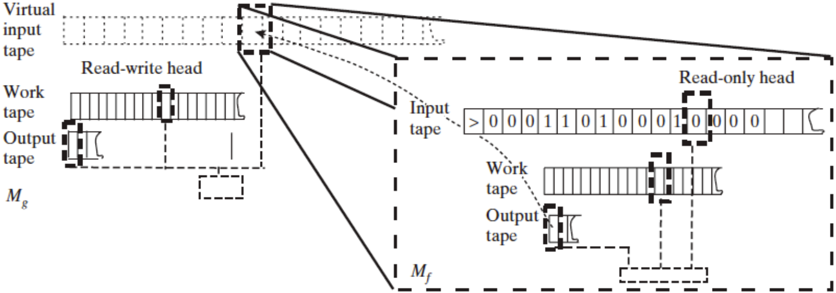
Note: Logspace NDTM has a read-once certificate tape

Lemma 19

If f,g are implicitly logspace computable, then so is $h(x) = g(f(x))$.

Proof:

Basic idea: To compute bits $h(x)_j = g(f(x))_j$, (re-)compute bits $f(x)_j$ used by g on-the-fly and on a "virtual tape" of M_g (i.e., special portion of M_g's tape).



Lemma 20

1. *If* $B \leq_l C$ *and* $C \leq_l D$*, then* $B \leq_l D$*.*
2. *If* $B \leq_l C$ *and* $C \in L$*, then* $B \in L$*.*

Proof:

1. Apply Lemma 19 to $f = B \leq_l C$ and $g = C \leq_l D$.
2. Apply Lemma 19 to $f = B \leq_l C$ and $g = M_C$.

Lemma 21

PATH is NL-complete

Proof:

$PATH \in NL$ (Theorem 5). For $L \in NL$, implicitly represent $M_L(x)$

- Node $C \in G_{M_L,x}$ takes log-space
- $(C, C') \in E_{G_{M_L,x}}$ checked in log-space: Run M_L from C to see if C' reached by some non-deterministic choice $\square$

Theorem 22 (Immerman-Szelepcsényi theorem)

$\overline{PATH} \in NL$

Corollary 1

If $S(n) \geq \log n$ *is space constructible,*

$$NSPACE(S(n)) \subseteq coNSPACE(S(n))$$

Corollary 2

$NL = coNL$

$$L \subseteq NL \subseteq P \subseteq NP \subseteq PSPACE \subseteq EXP$$

Note: We know $L \subset PSPACE$ and $P \subset EXP$, but we don't know which inclusions are strict :-(