Space Complexity

Space is a computation resource. Unlike time it can be reused.

Synopsis

1. Space Bounded Computation
2. Logspace Reduction
3. PSPACE Completeness
4. Savitch Theorem
5. NL Completeness
6. Immerman-Szelepcsényi Theorem

Space Bounded Computation

Space Bounded Computation

Let $S : \mathbf{N} \to \mathbf{N}$ and $L \subseteq \{0,1\}^*$.

We say that $L \in \mathbf{SPACE}(S(n))$ if there is some c and some TM deciding L that never uses more than $cS(n)$ nonblank worktape locations on inputs of length n.

Space Constructible Function

Suppose $S : \mathbf{N} \to \mathbf{N}$ and $S(n) \geq \log(n)$.

1. S is space constructible if there is a Turing Machine that computes the function $1^n \mapsto \llcorner S(n) \lrcorner$ in $O(S(n))$ space.
2. S is fully space constructible if there is a Turing Machine that upon receiving 1^n uses exactly $S(n)$-space.

The two definitions are equivalent in terms of marking cells.

Space Bounded Computation, the Nondeterministic Case

$L \in \mathbf{NSPACE}(S(n))$ if there is some c and some NDTM deciding L that never uses more than $cS(n)$ nonblank worktape locations on inputs of length n, regardless of its nondeterministic choices.

For space constructible function $S(n)$ we could allow a machine in $\mathbf{NSPACE}(S(n))$ to diverge and to use more than $cS(n)$ space in unsuccessful computation paths.

Configuration

A configuration of a running TM $\mathbb{M}$ with input x consists of the following:

- the state;
- the content of the work tape; [In the study of space complexity one may always assume that there is one work tape.]
- the head positions.

We write $C_{\mathtt{start}}$ for the unique initial configuration.

We assume that there is a single accepting configuration $C_{\mathtt{accept}}$.

Configuration Graph

A configuration graph $G_{\mathbb{M},x}$ of $\mathbb{M}$ with input x is a directed graph:

- the nodes are configurations;
- the arrows are one-step computations.

"$\mathbb{M}$ accepts x" iff "there is a path in $G_{\mathbb{M},x}$ from $C_{\mathtt{start}}$ to $C_{\mathtt{accept}}$".

Reachability Predicate for Configuration Graph

Suppose $\mathbb{M}$ is an $S(n)$-space TM.

- A vertex of $G_{\mathbb{M},x}$ is described using $O(S(|x|))$ bits.
- Therefore $G_{\mathbb{M},x}$ has at most $2^{O(S(|x|))}$ nodes.

There is an $O(S(n))$-size (string) CNF $\varphi_{\mathbb{M},x}$ such that for every two configurations C and C',

- $\varphi_{\mathbb{M},x}(C, C') = 1$ iff $C \to C'$ is an edge in $G_{\mathbb{M},x}$.

$\varphi_{\mathbb{M},x}(C, C')$ can be checked by essentially comparing C and C' bit by bit, accomplished in both

- $O(S(n))$ time, and $O(\log S(n))$ space.

Space vs. Time

Theorem. Suppose $S(n) : \mathbf{N} \to \mathbf{N}$ is space constructible. Then

$$\mathbf{TIME}(S(n)) \subseteq \mathbf{SPACE}(S(n)) \subseteq \mathbf{NSPACE}(S(n)) \subseteq \mathbf{TIME}(2^{O(S(n))}).$$

A TM for $\mathbf{NSPACE}(S(n)) \subseteq \mathbf{TIME}(2^{O(S(n))})$ constructs $G_{\mathbb{M},x}$ in $2^{O(S(n))}$ time, and then applies the breadth first search algorithm to the reachability instance

$$\langle G_{\mathbb{M},x}, C_{\texttt{start}}, C_{\texttt{accept}} \rangle.$$

Space vs. Time

Theorem. For all space constructible $S(n)$, $\mathbf{TIME}(S(n)) \subseteq \mathbf{SPACE}(S(n)/\log S(n))$.

1. Hopcroft, Paul and Valiant. On Time versus Space and Related Problems. FOCS, 1975.

Space Complexity Class

$$
\begin{aligned}
\mathbf{L} &\stackrel{\text{def}}{=} \mathbf{SPACE}(\log(n)), \\
\mathbf{NL} &\stackrel{\text{def}}{=} \mathbf{NSPACE}(\log(n)), \\
\mathbf{PSPACE} &\stackrel{\text{def}}{=} \bigcup_{c>0} \mathbf{SPACE}(n^c), \\
\mathbf{NPSPACE} &\stackrel{\text{def}}{=} \bigcup_{c>0} \mathbf{NSPACE}(n^c).
\end{aligned}
$$

Games are Harder than Puzzles

$$\mathbf{NP} \subseteq \mathbf{PSPACE}.$$

Example

The following problems are in $\mathbf{L}$:

$$
\begin{aligned}
\texttt{EVEN} &\stackrel{\text{def}}{=} \{x \mid x \text{ has an even number of } 1's\}, \\
\texttt{PLUS} &\stackrel{\text{def}}{=} \{(\llcorner m \lrcorner, \llcorner n \lrcorner, \llcorner m+n \lrcorner) \mid m, n \in \mathbf{N}\}, \\
\texttt{MULP} &\stackrel{\text{def}}{=} \{(\llcorner m \lrcorner, \llcorner n \lrcorner, \llcorner m \times n \lrcorner) \mid m, n \in \mathbf{N}\}.
\end{aligned}
$$

PATH is in $\mathbf{NL}$

$$\mathtt{PATH} = \{\langle G, s, t\rangle \mid \text{there is a path from } s \text{ to } t \text{ in the digraph } G\}.$$

Theorem. $\mathtt{PATH} \in \mathbf{NL}$.

Proof.

Both a node and a counter can be stored in logspace. $\square$

Universal Turing Machine without Space Overhead

Theorem. There is a universal TM that operates without space overhead for input TM's with space complexity $\geq \log(n)$.

A universal TM can simulate $\mathbb{M}_\alpha$ by recording all the non-blank tape content of $\mathbb{M}_\alpha$ in its single work tape, and k counters are used to store the locations of the readers.

Some additional space, whose size depends only on $\mathbb{M}_\alpha$, is necessary for bookkeeping.

Space Hierarchy Theorem

Theorem. If f, g are space constructible such that $f(n) = o(g(n))$, then

$$\mathbf{SPACE}(f(n)) \subsetneq \mathbf{SPACE}(g(n)).$$

We design $\mathbb{V}$ by modifying the universal machine so that

- $\mathbb{V}(x)$ simulates $\mathbb{M}_x(x)$, and
- it stops when it is required to use more than $g(n)$ space, and
- it negates the result after it completes simulation.

If $\mathbb{V}$ was executed in $f(n)$ space, then $\mathbb{V} = \mathbb{M}_\alpha$ for some large enough α so that $\mathbb{V}$ can complete the simulation of $\mathbb{M}_\alpha$ on α.

But then $\overline{\mathbb{M}_\alpha(\alpha)} = \mathbb{V}(\alpha) = \mathbb{M}_\alpha(\alpha)$.

---

1. J. Hartmanis and R. Stearns. On the Computational Complexity of Algorithms. Transactions of AMS, 117:285-306, 1965.

Logspace Reduction

Logspace Reduction

A function $f: \{0,1\}^* \to \{0,1\}^*$ is implicitly logspace computable if the following hold:

1. $\exists c. \forall x. |f(x)| \leq c|x|^c$,
2. $\{\langle x, i \rangle \mid i \leq |f(x)|\} \in \mathbf{L}$ and
3. $\{\langle x, i \rangle \mid f(x)_i = 1\} \in \mathbf{L}$.

Problem B is logspace reducible to problem C, written $B \leq_L C$, if there is an implicitly logspace computable f such that $x \in B$ iff $f(x) \in C$.

- Logspace reductions are Karp reductions. [The converse implication is unknown.]
- All known NP-completeness results can be established using logspace reduction.

Transitivity of Logspace Reduction

Lemma. If $B \leq_L C$ and $C \leq_L D$ then $B \leq_L D$.

Let $\mathbb{M}_f, \mathbb{M}_g$ be logspace machines that compute $x, i \mapsto f(x)_i$ respectively $y, j \mapsto g(y)_j$.
We construct a machine that, given input x, j with $j \leq |g(f(x))|$, outputs $g(f(x))_j$.

- The machine operates as if $f(x)$ were stored on a virtual tape.
  - It stores the address i of the current cell of the virtual tape.
  - It uses $O(\log|f(x)|) = O(\log|x|)$ space to calculate $g(f(x))_j$.

Logspace Computability

A function $f\colon \{0,1\}^* \to \{0,1\}^*$ is logspace computable if it can be computed by a TM that has a write-once output tape using $O(\log n)$ work tape space.

Lemma. Implicitly logspace computability $=$ logspace computability.

PSPACE Completeness

Space Completeness

A language L' is $\mathbf{PSPACE}$-hard if $L \leq_L L'$ for every $L \in \mathbf{PSPACE}$.

If in addition $L' \in \mathbf{PSPACE}$ then L' is $\mathbf{PSPACE}$-complete.

Quantified Boolean Formula

A quantified Boolean formula (QBF) is a formula of the form

$$Q_1 x_1 \, Q_2 x_2 \ldots Q_n x_x . \varphi(x_1, \ldots, x_n)$$

where each Q_i is one of the two quantifiers $\forall, \exists$, $x_1, \ldots, x_n$ range over $\{0, 1\}$, and φ is a quantifier free Boolean formula containing no free variables other than $x_1, \ldots, x_n$.

Let `TQBF` be the set of true QBFs.

Quantified Boolean Formula

Suppose φ is a CNF.

- $\varphi \in \mathtt{SAT}$ if and only if $\exists \widetilde{x}.\varphi \in \mathtt{TQBF}$.
- $\varphi \in \mathtt{TAUTOLOGY}$ if and only if $\forall \widetilde{x}.\varphi \in \mathtt{TQBF}$.

Stockmeyer-Meyer Theorem. $\mathtt{TQBF}$ is $\mathbf{PSPACE}$-complete.

---

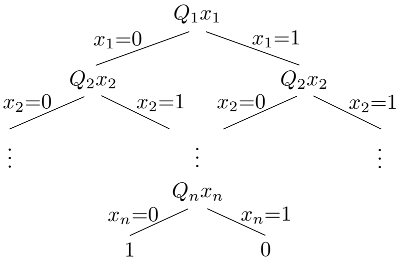
1. Larry Stockmeyer, Albert Meyer. Word Problems Requiring Exponential Time. STOC, 1973.

Proof of Stockmeyer-Meyer Theorem

TQBF $\in$ **PSPACE**. Suppose $\psi = Q_1x_1Q_2x_2\ldots Q_nx_n.\varphi(x_1,\ldots,x_n)$.

- A counter of length n is identified to an assignment.
- Apply the depth first tree traversal algorithm.

It is actually a linear space algorithm.



Proof of Stockmeyer-Meyer Theorem

Let $\mathbb{M}$ be a TM that decides L in polynomial space, say $S(n)$ space.

We reduce $x \in \{0,1\}^*$ to a QBF φ_x of size $O(S(|x|)^2)$ in logspace such that $\mathbb{M}(x) = 1$ iff φ_x is true.

---

1. Construct in logspace ψ_0 such that $\psi_0(C, C')$ is true if and only if $C \to C'$.

2. Let $\psi_i(C, C')$ be true if and only if there exists a path of length $\leq 2^i$ from C to C'. It can be defined by the following formula, which is computable in logspace.

$$\exists C'' \forall D^1 \forall D^2 . ((D^1 = C \wedge D^2 = C'') \vee (D^1 = C'' \wedge D^2 = C')) \Rightarrow \psi_{i-1}(D^1, D^2).$$

3. Now $|\psi_i| = |\psi_{i-1}| + c \cdot S(|x|)$. Hence $|\varphi_x| = |\psi_{S(|x|)}| = O(S(|x|)^2)$.

Every variable is coded up by a string of length $\log S(|x|)$. But ...

QBF Game

Two players make alternating moves on a board of the game.

$$\exists x_1 \forall x_2 \exists x_3 \forall x_4 \cdots \exists x_{2n-1} \forall x_{2n}.\varphi(x_1, \ldots, x_{2n}).$$

Player I moves first. It has a winning strategy if φ is true after Player II's last move, no matter how Player II plays.

- Deciding if Player I has a winning strategy for QBF game is $\mathbf{PSPACE}$-complete.

1. Christos Papadimitriou. Games Against Nature. FOCS, 1983.

Savitch Theorem

Savitch Theorem. If S is space constructible then $\mathbf{NSPACE}(S(n)) \subseteq \mathbf{SPACE}(S(n)^2)$.

Suppose $\mathbb{N}$ is an NDTM that decides L in $S(n)$ space.

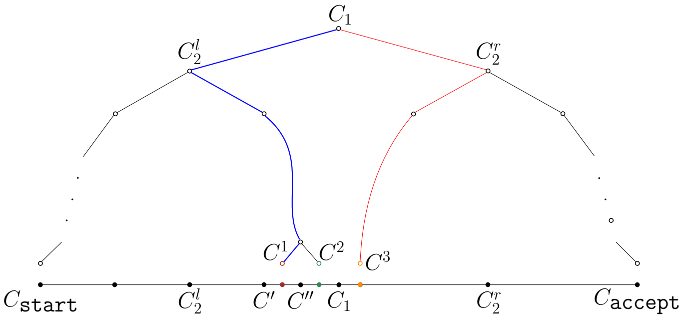
Given $x \in \{0,1\}^*$, a divide-and-conquer depth first algorithm can be designed that searches for a path from $C_{\mathtt{start}}$ to $C_{\mathtt{accept}}$ in $G_{\mathbb{N},x}$.

The depth of the recursive calls is $S(|x|)$.

---

1. W. Savitch. Relationships between Nondeterministic and Deterministic Tape Complexities. JCSS, 177-192, 1970.

Proof of Savitch Theorem



$$\mathbf{PSPACE} = \mathbf{NPSPACE}$$

NL Completeness

$\mathbf{NL}$-Completeness

C is $\mathbf{NL}$-complete if it is in $\mathbf{NL}$ and $B \leq_L C$ for every $B \in \mathbf{NL}$.

$\mathbf{NL}$-Completeness

Theorem. PATH is $\mathbf{NL}$-complete.

Suppose a nondeterministic TM $\mathbb{N}$ decides L in $O(\log(n))$ space. A logspace reduction from L to PATH is defined by the following reduction:

$$x \mapsto \langle G_{\mathbb{N},x}, C_{\mathtt{start}}, C_{\mathtt{accept}} \rangle.$$

The graph $G_{\mathbb{N},x}$ is represented by an adjacent matrix, every bit of it can be calculated in $O(|C|) = O(\log |x|)$ space.

We may assume that all space bounded TM's terminate by making use of counters. It follows that PATH remains $\mathbf{NL}$-complete if only acyclic graphs are admitted.

$\mathbf{NL}$ is nothing but PATH.

Immerman-Szelepcsényi Theorem

Savitch Theorem implies $\mathbf{coNPSPACE} = \mathbf{NPSPACE}$.

However $\mathbf{coNL} = \mathbf{NL}$ is a different story.

Immerman-Szelepcsényi Theorem. $\overline{\texttt{PATH}} \in \mathbf{NL}$.

1. R. Szelepcsényi. The Method of Forcing for Nondeterministic Automata. Bulletin of EATCS, 1987.
2. N. Immerman. Nondeterministic Space is Closed under Complementation. SIAM Journal Computing, 1988.

Proof of Immerman-Szelepcsényi Theorem

Design a logspace NDTM $\mathbb{N}$ such that for vertices s, t of a graph G with n-vertices, $\mathbb{N}(\langle G, s, t\rangle) = 1$ iff there is no path from s to t.

For $i \in [n-1]$,

- let C_i be the set of nodes reachable from s in i-steps, and [By definition $C_1 \subseteq C_2 \subseteq \ldots \subseteq C_{n-1}$.]
- let $c_i = |C_i|$. Notice that c_1 can be stored in logspace.

We can store a fixed number of c_i's, but we cannot store any of C_i's.

Set $c_{i+1} = 0$. For each vertex $v \neq s$, $\mathbb{N}$ guesses C_i and increments c_{i+1} if either $v \in C_i$ or $u \to v$ for some $u \in C_i$.

- For each u in C_i check that s reaches to u in i steps. This is `PATH`.
- A counter is maintained to ensure that $|C_i| = c_i$.

After c_{n-1} has been calculated, guess C_{n-1} and accept if $t \notin C_{n-1}$.

Immerman-Szelepcsényi Theorem

Corollary. For every space constructible $S(n) \geq \log(n)$, one has

$$\mathbf{coNSPACE}(S(n)) = \mathbf{NSPACE}(S(n)).$$

Theorem. $2\mathtt{SAT}$ is $\mathbf{NL}$-complete.

Given $\langle G, s, t\rangle$, where G is acyclic, we translate an edge $x \rightarrow y$ to the clause $\overline{x} \vee y$. We also add clauses s and $\overline{t}$. This is a logspace reduction from acyclic $\overline{\mathtt{PATH}}$ to $2\mathtt{SAT}$.

There is also a logspace reduction from $2\mathtt{SAT}$ to $\overline{\mathtt{PATH}}$.

By Hierarchy Theorems some of the following inclusions are strict.

$$\mathbf{L} \subseteq \mathbf{NL} \subseteq \mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{PSPACE} \subseteq \mathbf{EXP}.$$

Yet we don't know which is strict.

- It is widely believed that $\mathbf{NL} \subsetneq \mathbf{P}$.
- "$\mathbf{L} \stackrel{?}{=} \mathbf{NL}$" is a major open problem in the structural theory.