

CSCI 1590
Intro to Computational
Complexity
Space Complexity

John E. Savage

Brown University

February 11, 2008

- 1 Complexity Classes
- 2 Proper Resource Bounds
- 3 Hierarchy Theorems
- 4 Savitch's Theorem

Overview

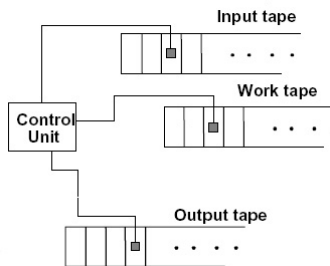
- Today's lecture deals with space-bounded complexity classes.
- A principal objective of today's lecture is to prove Savitch's Theorem, i.e. **NSPACE** $(r(n))$, the set of languages recognized in non-deterministic space $r(n)$, is contained in **SPACE** $(r^2(n))$.

Complexity Classes

- Complexity classes are defined in terms of resource bounds.
- Examples: logarithms, polynomials of logs, polynomials, and exponentials.
- Avoid functions that cannot be computed in the time and/or space they define.

Space on Turing Machines

- Input tape is read-only.
- Output tape is write-only.
- Space is maximum number of cells used on work tape.



Proper Resource Bounds

Definition

A function $r : N \mapsto N$ is **proper** if it is non-decreasing and for letter a there is a deterministic multi-tape TM M that on all inputs of length n in time $O(n + r(n))$ and space $O(r(n))$ writes the string $a^{r(n)}$ on one of its tapes and halts.

- If $r(n)$ is proper, a DTM, M_r , exists that can limit a computation on an input of length n to time $O(n + r(n))$ and space $O(r(n))$.

Assumptions

- We assume that resource bounds are proper but otherwise do not make use of this fact.
- That is, space and time are defined for DTMs (NTMs) using proper resource functions.

Space and Time Complexity Classes

- $\text{TIME}(r(n))$ and $\text{SPACE}(r(n))$ are sets of languages accepted in $r(n)$ time and space by DTMs, $n = \text{length of input}$.
- $\text{NTIME}(r(n))$ and $\text{NSPACE}(r(n))$ are sets of languages accepted in $r(n)$ time and space by NTMs, $n = \text{length of input}$.

$$\mathbf{P} = \bigcup_k \mathbf{TIME}(n^k) \qquad \mathbf{NP} = \bigcup_k \mathbf{NTIME}(n^k)$$

$$\mathbf{EXPTIME} = \bigcup_k \mathbf{TIME}(2^{n^k}) \qquad \mathbf{NEXPTIME} = \bigcup_k \mathbf{NTIME}(2^{n^k})$$

Hierarchy Theorems

Theorem (Time Hierarchy Theorem)

If $r(n) \geq n$ is a proper resource function, then **TIME**($r(n)$) is strictly contained in **TIME**($r(n) \log r(n)$).

Theorem (Space Hierarchy Theorem)

If $r(n)$ and $s(n)$ are proper resource functions and $r(n) = o(s(n))$, then **SPACE**($r(n)$) is strictly contained in **SPACE**($s(n)$).

Theorem

P \subseteq **NP** \subseteq **EXPTIME** \subseteq **NEXPTIME** but **P** is strictly contained in **EXPTIME**. Also, **NP** is strictly contained in **NEXPTIME**.

Configuration Graphs

Definition

A **configuration graph** of a Turing machine M is a graph in which the vertices are configurations of M on input word w and in which edges join two configurations if it is possible to move from one configuration to another.

For DTMs configuration graphs have one edge directed from one configuration to another. However, for NTMs, these graphs may have more than one edge between configurations.

REACHABILITY

REACHABILITY

Instance: A directed graph $G = (V, E)$ and a pair of vertices $u, v \in V$.

Answer: “Yes” if there is a directed path in G from u to v .

Theorem (Savitch)

REACHABILITY is in **SPACE** $(\log^2 n)$, $n = |V|$.

Why REACHABILITY?

Given a TM M and input w , construct configuration graph with one vertex for each configuration. If a final state is “reachable,” w is accepted by M .

Consequence

Nondeterminism doesn't increase the power of TMs when restricted to using an amount of space polynomial in the length of the input!

Space-Bounded Complexity Classes

Class	Space	By	Note
L	Logarithmic	DTM	
NL	Logarithmic	NDTM	$L \subseteq NL$
L^2	Square Log	DTM	
PSPACE	Polynomial	DTM	
NPSPACE	Polynomial	NTM	$PSPACE \subseteq NPSPACE$

First REACHABILITY Algorithm

- REACHABILITY can be solved using depth first search in $T = O(|E|)$ steps on a RAM.
- Can simulate T -step RAM program on DTM in $O(T^3)$ steps. Thus, DFS can be realized in $O(|E|^3)$ steps.
- Unfortunately, the space used for DFS on both machines is polynomial in $|V|$ and $|E|$.
- Savitch's Theorem shows reachability can be realized in deterministic space $O(\log^2 |V|)$, a big improvement.
- We use this fact to show **PSPACE = NPSPACE**.

Savitch's Theorem

Theorem

REACHABILITY is in **SPACE** $(\log^2 n)$, $n = |V|$.

Proof

Let $m = \log_2 n$. For vertices $a, b \in V$ and $k \leq m$, let $\text{PATH}(a, b, 2^k) = 1$ if there is a path between a and b of length $\leq 2^k$ and 0 otherwise. A "Yes" instance (G, u, v) of REACHABILITY has $\text{PATH}(u, v, 2^m) = 1$.

$\text{PATH}(a, b, 2^k) = 1$ if and only if there exists a vertex z such that $\text{PATH}(a, z, 2^{k-1}) = 1$ and $\text{PATH}(z, b, 2^{k-1}) = 1$. Thus, we evaluate $\text{PATH}(a, b, 2^k)$ recursively.

We give an $O(\log^2 |V|)$ -space DTM for $\text{PATH}(u, v, 2^m)$.

Proof of Savitch's Theorem

Construct an algorithm to compute $\text{PATH}(a, b, 2^k)$ using
 $\text{PATH}(a, b, 2^k) = \text{PATH}(a, z, 2^{k-1}) \wedge \text{PATH}(z, b, 2^{k-1})$

- $\text{PATH}(a, b, 1) = 1$ if $(a, b) \in E$, $\text{PATH}(a, b, 1) = 0$ otherwise.
- For all z until $\text{PATH}(a, b, 2^k) = 1$ or exhaust all z 's, recursively evaluate $\text{PATH}(a, z, 2^{k-1})$. If $\text{PATH}(a, z, 2^{k-1})$ is 1, evaluate $\text{PATH}(z, b, 2^{k-1})$. If also 1, $\text{PATH}(a, b, 2^k)$ is also 1.
- To implement recursion, push **activation records (ARs)** of form (a, b, k) on a stack. An AR specifies start and end vertices and path length 2^k .
- Goal: Determine if $\text{PATH}(u, v, 2^m) = 1$, where n vertices and $m = \lceil \log_2 n \rceil$.

Sketch of DTM for Savitch's Algorithm

- DTM has input and two work tapes
- Instance of REACHABILITY (graph G and vertices u and v) is on input tape.
- 1st work tape holds stack of ARs (a, b, k) .
- Initialize stack with $(u, v, \log_2 n)$.
- 2nd work tape used to evaluate AR.

Sketch of DTM for Savitch's Algorithm

- Three kinds of AR: a) complete, b) initial segment, c) final segment
- Start with complete AR (a, b, k) on 1st tape (stack).
- Choose first z . Form initial segment $(a, z, k - 1)$. Push AR onto stack.
- If $\text{PATH}(a, z, 2^{k-1}) = 1$, evaluate $\text{PATH}(z, b, 2^{k-1})$ (final segment).
 - (Determine $(z, b, k - 1)$ from ARs $(a, z, k - 1)$ and (a, b, k) , which are on stack.)
 - If $\text{PATH}(z, b, 2^{k-1}) = 1$, done.
 - If not, pick next z , if possible
- If $\text{PATH}(a, z, 2^{k-1}) = 0$, try another z .
- Evaluate $\text{PATH}(a, z, 2^0)$ from graph G .

Space Used by Savitch's Algorithm

- 2^{nd} work tape only used to evaluate elementary ARs $(a, b, 0)$.
- 1^{st} work tape contains at most $O(\log n)$ ARs because depth of recursion is $O(\log n)$.
- Each AR (a, b, m) uses $O(\log n)$ space (cells) because a and b are one of n vertices and $\lceil \log_2 n \rceil$ bits are needed to represent them. m can be represented with $\lceil \log_2 \lceil \log_2 n \rceil \rceil$ bits.
- Total space used is $O(\log^2 n)$.

Application of Savitch's Theorem

Corollary

Let $r(n)$ be proper and $r(n) = \Omega(\log n)$. Then,
NSPACE $(r(n)) \subseteq$ **SPACE** $(r^2(n))$.

Proof

Let $L \in$ **NSPACE** $(r(n))$. There is a NTM, M , recognizing L . It's configuration is a tuple $(p, h_1, \dots, h_{s+2}, x_1, \dots, x_{s+2})$ where p is M 's state, h_j is position of head on j th tape, x_j is nonblank contents of j th tape. One state for each configuration, an edge exists between configurations if M can move from one to other (deterministically or nondeterministically).

Application of Savitch's Theorem

Proof (cont.)

Configuration graph $G(M, w)$ on input w has path from start configuration to accepting one if and only if M accepts w . Number of vertices in $G(M, w)$ when $n = |w|$ is $C = |Q|(nr(n)^s)c^{(s+2)r(n)}$.

Space = $O(\log^2 C) = O(r^2(n))$.

Comment

It is important to note that the entire configuration does not have to be constructed. Instead, it suffices to determine for two configurations whether one is reachable in one step from the other.

Relationships Between Complexity Classes

- **$NL \subseteq L^2 \subseteq PSPACE$**
- **L^2** strictly contained within **$PSPACE$** by space hierarchy theorem
- **$L \subseteq NL \subseteq P \subseteq PSPACE \subseteq EXPTIME \subseteq NEXPTIME$**
 - See pages 341-343 of Savage textbook.