Complexity Theory

Georg Moser

Institute of Computer Science @ UIBK

Summer 2008

Outline

- Summary of Last Lecture: Crossing Sequences
- Exercises
- Linear Speed-Up
- Savitch's Theorem

Theorem

If M runs in $o(\log\log n)$ space, then M accepts a regular set

Fact

$\exists$ a non-regular set accepted in $O(\log\log n)$ space

$$\{\#b_k(0)\#b_k(1)\#b_k(2)\#\dots\#b_k(2^k-1)\# \mid k \geqslant 0\}$$

$b_k(n)$ is the k-bit (binary) representation of n

Proof

using similar crossing argument, but the following lemma

Lemma

If here is a fixed finite bound k on the amount of space used by M on accepted inputs, then $L(M)$ is a regular set

∎

Time- and Space-Boundedness

let $T\colon \mathbb{N} \to \mathbb{N}$ and $S\colon \mathbb{N} \to \mathbb{N}$ be numeric functions; as usual we write $\log n$ for $\lceil \log n \rceil$, etc.

Definition — time-bounded

- nondeterministic TM runs in time $T(n)$ or
- TM is $T(n)$ time-bounded
- if on all but finitely many inputs x, no computation path takes more than $T(|x|)$ steps before halting

Definition — space-bounded

- nondeterministic TM runs in space $S(n)$ or
- TM is $S(n)$ space-bounded
- if on all but finitely many inputs x, no computation path uses more than $S(|x|)$ worktape cells

Linear Speedup

$$\text{DTIME}(T(n)) := \{\mathrm{L}(M) \mid M \text{ is a deterministic multitape TM running in time } T(n)\}$$
$$\text{NTIME}(T(n)) := \{\mathrm{L}(M) \mid M \text{ is a nondeterministic multitape TM running in time } T(n)\}$$
$$\text{DSPACE}(S(n)) := \{\mathrm{L}(M) \mid M \text{ is a deterministic multitape TM running in space } S(n)\}$$
$$\text{NSPACE}(S(n)) := \{\mathrm{L}(M) \mid M \text{ is a nondeterministic multitape TM running in space } S(n)\}$$

Theorem (linear speed-up)

let $T(n) \geqslant n + 1$, $S(n) \geqslant \Omega(\log n)$; for any $c \geqslant 1$:

$$\text{DTIME}(cT(n)) \subseteq \text{DTIME}(T(n)) \qquad \text{NTIME}(cT(n)) \subseteq \text{NTIME}(T(n))$$
$$\text{DSPACE}(cS(n)) \subseteq \text{DSPACE}(S(n)) \qquad \text{NSPACE}(cS(n)) \subseteq \text{NSPACE}(S(n))$$

Common Complexity Classes

$$\text{LOGSPACE} := \text{DSPACE}(\log n) \qquad \text{NPSPACE} := \text{NSPACE}(n^{\mathrm{O}(1)})$$
$$\text{NLOGSPACE} := \text{NSPACE}(\log n) \qquad \text{EXPTIME} := \text{DTIME}(2^{n^{\mathrm{O}(1)}})$$
$$\text{P} := \text{DTIME}(n^{\mathrm{O}(1)}) \qquad \text{NEXPTIME} := \text{NTIME}(2^{n^{\mathrm{O}(1)}})$$
$$\text{NP} := \text{NTIME}(n^{\mathrm{O}(1)}) \qquad \text{EXPSPACE} := \text{DSPACE}(2^{n^{\mathrm{O}(1)}})$$
$$\text{PSPACE} := \text{DSPACE}(n^{\mathrm{O}(1)}) \qquad \text{NEXPSPACE} := \text{NSPACE}(2^{n^{\mathrm{O}(1)}})$$

we abbreviate $\bigcup_{k>0} \text{DTIME}(n^k)$ by $\text{DTIME}(n^{\mathrm{O}(1)})$;
similarly for other classes

Basic Inclusions

$$\text{DTIME}(T(n)) \subseteq \text{NTIME}(T(n)) \qquad \text{DSPACE}(S(n)) \subseteq \text{NSPACE}(S(n))$$

Theorem 1

assume $S(n) \geqslant \log n$, then

$$
\begin{aligned}
①&: \mathrm{DTIME}(T(n)) \subseteq \mathrm{DSPACE}(T(n)) \\
②&: \mathrm{NTIME}(T(n)) \subseteq \mathrm{NSPACE}(T(n)) \\
③&: \mathrm{DSPACE}(S(n)) \subseteq \mathrm{DTIME}(2^{\mathrm{O}(S(n))}) \\
④&: \mathrm{NSPACE}(S(n)) \subseteq \mathrm{NTIME}(2^{\mathrm{O}(S(n))})
\end{aligned}
$$

Proof

①,② follow as a TM can scan only one tape cell in every step

③,④ we show how to modify a given TM M running in space $S(n)$

- assume M features a single read-only input tape; a single worktape
- there are at most $qnS(n)d^{S(n)} =: c^{S(n)}$ configurations

  1. q — number of states
  2. n — cursor on the input tape
  3. $S(n)$ — cursor on the worktape
  4. d — cardinality of the alphabet

Proof (cont'd)

- any computation path taking more than $c^{S(n)}$-steps can be shortened
- on a new worktape install a timer counting up to $c^{S(n)}$
  1. no extra space, if use encoding in c-ary
  2. $\mathrm{O}(S(n))$ extra time for each step
- total time spent: $\mathrm{O}(S(n)) \cdot c^{S(n)} = 2^{\mathrm{O}(S(n))}$ ■

Theorem 2

assume $S(n) \geqslant \log n$, then

$$
\begin{aligned}
①&: \mathrm{NTIME}(T(n)) \subseteq \mathrm{DSPACE}(T(n)) \\
②&: \mathrm{NSPACE}(S(n)) \subseteq \mathrm{DTIME}(2^{\mathrm{O}(S(n))})
\end{aligned}
$$

Proof

① depth-first search on the computation tree

the obvious algorithm gives a $T(n)^2$-space bound

to remove the exponent ^{2} one wastes time: store only the choice-sequence and recomputes the configuration

Proof (cont'd)

② assume $S(n)$ is space-constructible

1. this means there exists a TM that can be used to mark off $S(n)$ worktape cells
2. write down all configurations of the nondeterministic machine, using at most $S(n)$ space
3. there are at most $c^{S(n)}$ of these, writable (all) in $S(n)c^{S(n)}$ time
4. we inductively mark all configurations reachable from the start configuration
5. this reachability argument works in quadratic time in the number of states
6. we obtain a $d^{S(n)}$-time bound

Proof (cont'd)

② we get rid of space-constructibility

1. the procedure above is iterated for

$$S = 0, 1, 2, \ldots$$

2. if in this process we need more space, we extend S
3. eventually we have to hit $S(n)$ and we can ignore anything beyond
4. the total time is a most

$$\sum_{S=0}^{S(n)} d^S \leqslant \frac{d^{S(n)+1} - 1}{d - 1} \ .$$

Savitch's Theorem

Theorem

let $S(n) \geqslant \log n$. Then

$$\mathrm{NSPACE}(S(n)) \subseteq \mathrm{DSPACE}(S(n)^2) \ ,$$

in particular PSPACE = NPSPACE