Complexity Theory

Lecture 11

Anuj Dawar

http://www.cl.cam.ac.uk/teaching/1819/Complexity

Savitch's Theorem

Further simulation results for nondeterministic space are obtained by other algorithms for Reachability.

We can show that Reachability can be solved by a *deterministic* algorithm in $O((\log n)^2)$ space.

Consider the following recursive algorithm for determining whether there is a path from a to b of length at most i.

$O((\log n)^2)$ space Reachability algorithm:

Path(a, b, i)
if $i = 1$ and $a \neq b$ and (a, b) is not an edge reject
else if (a, b) is an edge or $a = b$ accept
else, for each node x, check:

1. Path$(a, x, \lfloor i/2 \rfloor)$
2. Path$(x, b, \lceil i/2 \rceil)$

if such an x is found, then accept, else reject.

The maximum depth of recursion is $\log n$, and the number of bits of information kept at each stage is $3 \log n$.

Savitch's Theorem

The space efficient algorithm for reachability used on the configuration graph of a nondeterministic machine shows:

$$\mathsf{NSPACE}(f) \subseteq \mathsf{SPACE}(f^2)$$

for $f(n) \geq \log n$.

This yields

$$\mathsf{PSPACE} = \mathsf{NPSPACE} = \text{co-}\mathsf{NPSPACE}.$$

Complementation

A still more clever algorithm for Reachability has been used to show that nondeterministic space classes are closed under complementation:

If $f(n) \geq \log n$, then

$$\mathsf{NSPACE}(f) = \text{co-}\mathsf{NSPACE}(f)$$

In particular

$$\mathsf{NL} = \text{co-}\mathsf{NL}.$$

Logarithmic Space Reductions

We write

$$A \leq_L B$$

if there is a reduction f of A to B that is computable by a deterministic Turing machine using $O(\log n)$ workspace (with a *read-only* input tape and *write-only* output tape).

Note: We can compose $\leq_L$ reductions. So,

$$\text{if } A \leq_L B \text{ and } B \leq_L C \text{ then } A \leq_L C$$

NP-complete Problems

Analysing carefully the reductions we constructed in our proofs of NP-completeness, we can see that SAT and the various other NP-complete problems are actually complete under $\leq_L$ reductions.

Thus, if SAT $\leq_L A$ for some problem A in L then not only P $=$ NP but also L $=$ NP.

P-complete Problems

It makes little sense to talk of complete problems for the class P with respect to polynomial time reducibility $\leq_P$.

There are problems that are complete for P with respect to *logarithmic space* reductions $\leq_L$.
One example is CVP—the circuit value problem.

That is, for every language A in P,

$$A \leq_L \mathsf{CVP}$$

- If CVP $\in$ L then L $=$ P.
- If CVP $\in$ NL then NL $=$ P.

Reachability

Similarly, it can be shown that Reachability is, in fact, NL-complete.

For any language $A \in \mathsf{NL}$, we have $A \leq_L$ Reachability

$\mathsf{L} = \mathsf{NL}$ *if, and only if, Reachability* $\in \mathsf{L}$

Note: it is known that the reachability problem for *undirected* graphs is in L.

Provable Intractability

Our aim now is to show that there are languages (*or, equivalently, decision problems*) that we can prove are not in P.

This is done by showing that, for every *reasonable* function f, there is a language that is not in $\mathsf{TIME}(f)$.

The proof is based on the diagonal method, as in the proof of the undecidability of the halting problem.

Time Hierarchy Theorem

For any constructible function f, with $f(n) \geq n$, define the f-bounded *halting language* to be:

$$H_f = \{[M], x \mid M \text{ accepts } x \text{ in } f(|x|) \text{ steps}\}$$

where $[M]$ is a description of M in some fixed encoding scheme.
Then, we can show
$H_f \in \mathsf{TIME}(f(n)^2)$ and $H_f \notin \mathsf{TIME}(f(\lfloor n/2 \rfloor))$

Time Hierarchy Theorem
For any constructible function $f(n) \geq n$, $\mathsf{TIME}(f(n))$ is properly contained in $\mathsf{TIME}(f(2n+1)^2)$.