CMPT 710 - Complexity Theory: Week 3

Valentine Kabanets

September 27, 2005

1 Robust Time and Space Classes

"Robust" (intuitive notion): no reasonable changes to the model of computation should change the class; capable of classifying interesting problems.

Examples of robust classes:

1. L (contains Formula Value, integer multiplication and division),
2. P (contains circuit value, linear programming, max-flow),
3. PSPACE (contains 2-person games),
4. EXP (contains all of PSPACE).

2 Relationships among complexity classes

Theorem 1. *L $\subseteq$ P $\subseteq$ PSPACE $\subseteq$ EXP.*

The middle inclusion (P $\subseteq$ PSPACE) is trivial: in polynomial time, no TM can touch more than polynomial number of tape cells.

To prove the other two inclusions, we need the notion of a *configuration* of a TM. If at a given moment in time, a TM is in state q, and its tape contains symbols $\sigma_1\sigma_2\dots\sigma_i\sigma_{i+1}\dots\sigma_m$, and its tape head is scanning the symbol σ_{i+1}, then the configuration of the TM at this moment in time is

$$C = \sigma_1\sigma_2\dots\sigma_i q\sigma_{i+1}\dots\sigma_m.$$

(Note that the state name is immediately to the left of the symbol currently scanned by the TM.)

Start configuration (for a one-tape TM) on input x is $q_{start}x_1x_2\dots x_n$.

We say that a configuration C *yields* C' in one step if a TM in configuration C goes to configuration C' in one step, using its transition function.

We now define the *configuration graph* of a given TM M on input x:

- Nodes = configurations of M,
- Edges = $\{(C, C') \mid C \text{ yields } C' \text{ in one step}\}$

Easy Fact: The number of configurations of a 2-tape TM (with one read-only input tape and one work tape) is at most:

- n (input-tape head positions)
- $*f(n)$ (work-tape head positions)
- $*|Q|$ (state)
- $*|\Sigma|^{f(n)}$ (work-tape contents).

Note: the contents of the read-only input tape is not part of the configuration.

Proof of Theorem 1. For $f(n) = c \log n$, the number of configurations is at most

$$n * c \log n * c_0 * c_1^{c \log n} \leq n^{c_2},$$

a polynomial.

For $f(n) = n^c$, the number of configurations is at most

$$n * n^c * c_0 * c_1^{n^c} \leq 2^{n^{c_2}},$$

an exponential function.

To determine if a TM M accepts x,

1. construct the configuration graph of M on x;
2. check if q_{accept}-configuration is reachable from the start configuration (using, e.g., DFS); this can be done in time polynomial in the size of the graph.

Hence, for $f(n) = c \log n$, we need $\mathsf{poly}(n)$ time; and for $f(n) = n^c$, we need $\mathsf{poly}(2^{n^{c_2}})$ time. □

Theorem 2. $\mathsf{L} \subsetneq \mathsf{PSPACE}$ *and* $\mathsf{P} \subsetneq \mathsf{EXP}$.

Proof. We prove the first proper inclusion only; the second can be proved similarly. Note that $L \subseteq \mathsf{Space}(n)$ trivially. Now, by the Space Hierarchy Theorem, $\mathsf{Space}(n) \subsetneq \mathsf{Space}(n \log n)$. Finally, $\mathsf{Space}(n \log n) \subseteq \mathsf{PSPACE}$ trivially. □

As an immediate consequence, we obtain the following

Theorem 3. *Among the inclusions* $\mathsf{L} \subseteq \mathsf{P} \subseteq \mathsf{PSPACE}$ *at least one inclusion must be proper. Among the inclusions* $\mathsf{P} \subseteq \mathsf{PSPACE} \subseteq \mathsf{EXP}$ *at least one inclusion must be proper.*

Proof. Again we prove just the first part. Suppose that all inclusions are equalities. Then $\mathsf{L} = \mathsf{P} = \mathsf{PSPACE}$, contradicting Theorem 2. □

Major Open Questions:

- $\mathsf{L} \stackrel{?}{=} \mathsf{P}$
- $\mathsf{P} \stackrel{?}{=} \mathsf{PSPACE}$
- $\mathsf{PSPACE} \stackrel{?}{=} \mathsf{EXP}$

By Theorem 3, at least one of these three questions must have a negative answer. The bizarre fact is: *we do not know which one!* (even though we suspect that all of these questions have negative answers.)

3 "Padding" Technique

Suppose $L \in \mathsf{EXP}$ is decided by a TM M in time 2^{n^c}, for some constant c. Define a new language

$$L_{pad} = \{x\#^{2^{|x|^c}} \mid x \in L\}$$

(here, # is a new symbol outside the alphabet of M).

The TM M can be easily modified to M' that decides L_{pad}: just ignore the #'s. Now, the running time of M' on input $x\#^{2^{|x|^c}}$ of size $n = |x| + 2^{|x|^c}$ is $O(n)$, linear!

Why is this useful?

Theorem 4. *If* $\mathsf{P} = \mathsf{L}$*, then* $\mathsf{EXP} = \mathsf{PSPACE}$.

Proof. Take an arbitrary $L \in \mathsf{EXP}$. Construct L_{pad} as explained above. Since $L_{pad} \in \mathsf{P}$, by our assumption $\mathsf{P} = \mathsf{L}$, we get that $L_{pad} \in \mathsf{L}$. That is, there is some TM M_0 deciding L_{pad} in space $\log n$. This TM M_0 can be used to decide L as follows: On input x, simulate M_0 on the padded input $x\#^{2^{|x|^c}}$; accept iff M_0 accepts.

Now, the space we used on input x of size n is the space used by M_0 on the padded input $x\#^{2^{|x|^c}}$ of size $n + 2^{n^c}$, which is at most $\log(2^{n^c+1}) = n^c + 1$, a polynomial. Hence, we have that $L \in \mathsf{PSPACE}$.

$\square$

The theorem above is just one example of a more general phenomenon: "Padding" allows us to translate complexity statements upwards.

4 Nondeterminism

DTM (deterministic TM): next step of computation is completely determined by the current configuration of a TM

NTM (nondeterministic TM): there may be several possibilities for a next step

Formally. NTM's transition function is $\delta : Q \times \Sigma \to 2^{Q \times \Sigma \times \{L,R,-\}}$, i.e., a pair (state,symbol) is mapped to a set {(state,symbol,movement) }.

Thus, DTM's computation is a path; whereas NTM's computation is a tree.

Accept/Reject criteria for NTMs: $x \in L$ iff there exists some accepting computation.

Note: if $x \in L$, then there may be some rejecting computations as well.

Time $= \max_{\text{paths } p}$ {length of p }

Space $= \max_{\text{paths } p}$ {number of work-tape cells touched on a path p }

5 Nondeterministic Complexity Classes

$\mathsf{NTime}(f(n)) = \{L \mid$ some multi-tape NTM decides L in time at most $f(n)\}$

$\mathsf{NSpace}(f(n)) = \{L \mid$ some multi-tape NTM decides L in space at most $f(n)\}$

- $\mathsf{NP} = \cup_k \mathsf{NTime}(n^k)$

- $\mathsf{NEXP} = \cup_k \mathsf{NTime}(2^{n^k})$
- $\mathsf{NL} = \mathsf{NSpace}(\log n)$
- $\mathsf{NPSPACE} = \cup_k \mathsf{NSpace}(n^k)$

Remark We'll see later that $\mathsf{NPSPACE} = \mathsf{PSPACE}$, and so there is really no need to use the name NPSPACE.

6 Alternative definition of NP

$L \in \mathsf{NP}$ if there is a language $R \in \mathsf{P}$ and a constant c such that

$$L = \{x \mid \exists y, |y| \leq |x|^c, (x, y) \in R\}$$

Lemma 5. *The two given definitions of* NP *are equivalent.*

Proof. Exercise. Hint: y in Definition 2 = accepting path in Definition 1. □

7 Importance of NP

NTMs cannot be efficiently implemented. So they are an abstraction. But, a huge number of real-life problems are in NP because they are of the form: problem description such that a solution to the problem is "small" and the solution is "easy" to test for correctness. (So we can nondeterministically guess a solution, and then test its correctness in polytime.)

8 Relations

Lemma 6. $\mathsf{P} \subseteq \mathsf{NP}$ *and* $\mathsf{EXP} \subseteq \mathsf{NEXP}$.

Proof. Trivial: a DTM is a special case of an NTM. □

Theorem 7. $\mathsf{NP} \subseteq \mathsf{PSPACE}$

Proof. Try all possible y's (of polynomial length). □

Major Open Problem: $\mathsf{P} \stackrel{?}{=} \mathsf{NP}$

This is a problem of "Generating a solution vs. Recognizing a solution". Some examples: student vs. grader; composer vs. listener; writer vs. reader; mathematician vs. computer.