CS 6743 **Lecture 4**[1] **Fall 2007**

1 Robust Time and Space Classes

We say that a complexity class is "robust" (intuitive notion) if:

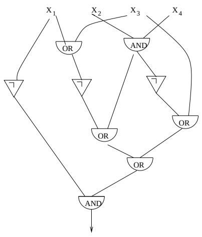
1. no reasonable changes to the model of computation changes the class, and
2. the class contains interesting problems.

For example, the class P is a robust class: changing the definition of a TM (say, to multi-tape TMs, or Random-Access Computer) will not change the class of polytime solvable problems. In contrast, the class of linear-time solvable problems is not robust! As we saw earlier, the language of palindromes is in linear time for 2-tape TMs, but not for 1-tape TMs.

To make it easier to think about these classes, think of complete problems for them.

- L (has reachability in forests as a complete problem)
- P (complete problem: boolean circuit value problem),
- PSPACE (evaluating a first-order formula),
- EXP (succinct circuit value problem).

Here, circuit value problem is defined as follows: consider a circuit (acyclic directed graph) with source nodes labeled by boolean input variables, sink node by a 0/1 value, and other nodes (gates) by either AND or OR or NOT. A NOT gate has one edge going into it, AND and OR at least two. The Circuit Value Problem (CVP) is to determine, given the values for input variables, the value at the output gate according to the labels of the internal gates.



In a succinct CVP the circuit description is given implicitly, e.g., as an algorithm that computes the description, so that the size of the actual circuit could be exponentially larger than the size of its description.

As we said earlier, we will equate "feasibly solvable" with polytime solvable. This means, of course, that for us the running time n^{10000} is feasible, which is not true in practice. However, there is an empirical fact, known as the *feasibility thesis*, stating that natural problems in the class P are usually solvable in small polynomial time, quadratic or better. This justifies our use of P as the class of feasible problems.

[1] This lecture is a modification of notes by Valentine Kabanets

2 Relationships among complexity classes

We'll show that any language decidable in space $s(n)$ can be decided in time, roughly, $2^{s(n)}$. This will yield the following.

Theorem 1. $\mathsf{L} \subseteq \mathsf{P} \subseteq \mathsf{PSPACE} \subseteq \mathsf{EXP}$.

The middle inclusion ($\mathsf{P} \subseteq \mathsf{PSPACE}$) 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 $\sigma_1\sigma_2\ldots\sigma_i\sigma_{i+1}\ldots\sigma_m$, and its tape head is scanning the symbol σ_{i+1}, then the configuration of the TM at this moment in time is

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

(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 $q_{start}x_1x_2\ldots x_n$.

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.

Example 1. Suppose that the tape contains the string "0110b" as its first $f(n) = 5$ cells, and that the head is pointing to the second occurrence of a 1, with the TM being in state q_3. The corresponding configuration is $01q_310b$.

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)
- $*|\Gamma|^{f(n)}$ (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. $\mathsf{L} \subsetneq \mathsf{PSPACE}$

Proof. 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 $L \in \mathsf{EXP}$, decided by some TM M in time 2^{n^c}. 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 $L \in \mathsf{PSPACE}$.

One technical point: If we actually were to write down the padded input $x\#^{2^{|x|^c}}$ on our work tape, this would make our space usage exponential in n. But, we do not need to write this padded string down! We know what it looks like to the right of x. So we can simulate M_0 on the virtual padded input $x\#^{2^{|x|^c}}$ by keeping a counter which tells us the position on the tape of M_0. If M_0 enquires about the position to the right of x, we respond with the symbol "#" (or the blank, if M_0 goes to the right of the entire padded input). Since the counter can assume the value at most 2^{n^c}, we need at most n^c bits for the counter. Thus, our new TM uses space at most polynomial in n. $\square$

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