CSC 2401, Introduction to computational complexity
Lecture #5, Wednesday, September 26, 2001
Lecturer: Allan Borodin
Scribe: George Katsirelos

1 Overview of previous results

In the previous lecture, we discussed the time and space hierarchy theorems. Specifically, the time hierarchy theorem states that

Theorem 1 *Let $t_2(n) \geq n$ be time constructible and*

$$\lim_{n\to\infty} \frac{t_1(n) log t_1(n)}{t_2(n)} = 0$$

Then

$$DTIME(t_1) \subsetneqq DTIME(t_2)$$

Similarly, the space hierarchy theorem states

Theorem 2 *Let $s_2(n) \geq n$ be time constructible and*

$$\lim_{n\to\infty} \frac{s_1(n)}{s_2(n)} = 0$$

Then

$$DSPACE(s_1) \subsetneqq DSPACE(s_2)$$

Simple observation: If $\mathcal{M}$ runs in space $s(n)$ and halts on every input, then $\mathcal{M}$ runs in time $2^{c \cdot s(n)}$ for some c. This is because it is going to halt in $|Q| \cdot n \cdot s(n) \cdot |\Gamma|^{s(n)}$ steps, as this is the maxiumum number of possible different configurations. Note that if $\mathcal{M}$ reaches the same configuration more than once, it will never halt.

2 Complexity classes hierarchy

So far, we have seen the following hierarchy of classes:

$$\mathcal{DL} \subseteq P \subseteq PSPACE \subseteq \cup_k DTIME(2^{n^k})$$

None of these inclusions are known to be proper. However, using the hierarchy theorems, we can see that

$\mathcal{DL} \subsetneqq PSPACE$

$P \subsetneqq \cup_c DTIME(2^{c \cdot n}) = EXP \subseteq \cup_k DTIME(2^{n^k})$

We can use diagonilization to prove $P \subsetneqq EXP$.
A surprising result that we can get is that

Theorem 3 *$EXP \neq PSPACE$*
Proof *(Sketch) Assume $EXP = PSPACE$. We assume some language $L' \in PSPACE$ and prove that this implies that $L \in PSPACE$. But $L' \in EXP \not\Rightarrow L \in EXP$. Since the 2 classes EXP, $PSPACE$ have different closure properties, they must be different. And now the complete proof.//*
Let $L \in DTIME(2^{n^2}) - EXP$. We know that L exists by the time hierarchy theorem. Now, let $L' = \{ \underbrace{1^{n^2}0x}_{n^2+n+1} : |x| = n, x \in L\}$. L' is a padded version of L. Obviously, since the input now has length $k = \mathcal{O}(n^2)$, $L' \in DTIME(2^k)$, therefore $L' \in EXP$. Since we have assumed that $EXP = PSPACE$, we also have $L' \in PSPACE$. Since L' is a padded version of L, L will run in polynomial space as well, so $L \in PSPACE$. However, by the assumption $EXP = PSPACE$, we also have $L \in EXP$, which is a contradiction, since we have assumed that $L \in DTIME(2^{n^2}) - EXP$ $\square$

Note that this proof cannot tell us if $EXP - PSPACE \neq \emptyset$ or $PSPACE - EXP \neq \emptyset$. We conjecture that both are true.

3 Non deterministic Turing Machines

So far, we have talked about deterministic Turing Machines. A non-deterministic Turing Machine $\mathcal{M}$ is a tuple

$$\mathcal{M} =< Q, \Sigma, \Gamma, \delta, F_1, F_2, q_0 >$$

which is similar to the definition of a deterministic TM, only the definition of δ is now:

$$(Q - F_1 - F_2) \times \Gamma^k \times \Sigma \rightarrow 2^{\{Q \times \Gamma^k \times \{L,R,S\}^{k+1}\}}$$

There is now a set of possible next steps, instead of a single next step. In other words, δ is now a relation:

$$\delta \subseteq (Q - F_1 - F_2) \times \Gamma^k \times \Sigma \times 2^{\{Q \times \Gamma^k \times \{L,R,S\}^{k+1}\}}$$

A non-deterministic turing machine o;pertaing on an input produces a computation tree (each path being a possible computationnsequence), instead of a a single computation sequence. Note that without loss of generality, the number of next possible steps can be as few as 2.

Definition: An input is said to be *accepted* by a nondeterministic TM, if there is a path down the tree that leads to an accepting state.
We will say that $\mathcal{M}$ runs in time $t(n)$ (respectively space s(n)), if
[$\mathcal{M}$ accepts $w \Rightarrow \exists$ accepting path with fewer than $t(|w|)$ steps] (respectively, [$\mathcal{M}$ accepts $w \Rightarrow \exists$ accepting path using less than $s(|w|)$ space.])

Fact: If $t(n) \geq n$ is time constructible (respectively $s(n) \geq \log n$ is space constructible) and $\mathcal{M}$ runs in time $t(n)$ (respectively $\mathcal{M}$ runs using space $s(n)$), then $\exists \mathcal{M}' : \mathcal{M}'$ always halts and every path using $O(t(n))$ steps (respectively using $O(s(n))$ space.)