CMSC 28100-1 / MATH 28100-1
Introduction to Complexity Theory
Fall 2017 – Homework 7
Solution

November 17, 2017

Exercise 1. Suppose $T\colon \mathbb{N} \to \mathbb{N}$ is a function with $T(n) \geqslant n+1$ such that there exists a *single tape* nondeterministic Turing machine M_T such that for every input x of length n, all branches of computation of M_T on x take time at most $T(n)$ (i.e., the machine M_T is $T(n)$-time bounded) and at least one branch of computation of M_T takes time exactly $T(n)$.

Show that a k-tape nondeterministic Turing machine running in a nondeterministic time $T(n)$ can be simulated by a 2-tape nondeterministic Turing machine running in nondeterministic time $O(T(n))$. Note that, unlike the case of deterministic time, there is no additional $\log(T(n))$ factor. Only state essential ideas (i.e., why we don't need the extra $\log(T(n))$ factor) in plain English like an algorithm.

Solution. Let N be the k-tape nondeterministic Turing machine to be simulated and let us describe the algorithm of the 2-tape Turing machine M that is going to simulate it.

The first tape of M has two tracks, with the input starting on the first track.

We start by copying the input x from the first track to the second track and rewind the head of the first tape to the beginning. Then we run M_T on the second track of the first tape (recall that M_T has a single tape), writing a mark on the second tape for every step that M_T takes. This means that by the end of the execution of M_T, the second tape has at most $T(|x|)$ marks and in at least one branch of computation, it has exactly $T(|x|)$ marks.

We then erase the second track of the first tape and transfer all marks from the second tape to the second track of the first tape.

Now we nondeterministically guess $m = T(|x|)$ transitions $t_1, \ldots, t_m$ of N writing them in the second tape (we count these transitions by looking at the marks in the second track of the first tape), and rewinding it by the end.

We then simulate each tape of N independently in the first tape of M according to $t_1, \ldots, t_m$. The first tape simulated starts with x (which is still in the first track of M) and all others start empty. In each simulation, we proceed according to $t_1, \ldots, t_m$ altering the first tape. If any guess t_i is wrong (i.e., the symbol read does not match the one we guessed we would read), we reject. If all simulations are correct and we halt in an accepting state, then we accept; otherwise, we reject.

Let us now show that $L(M) = L(N)$. It is easy to see that M can only accept words that N accepts. On the other hand, if N accepts a word x, then at least one branch of M correctly guesses the full accepting computation of N on x, so M accepts x.

Let us now analyze the time complexity of M.

Copying the input from the first track to the second track and rewinding takes time at most $2n+2$.

Running M_T takes time at most $T(n)$ and by the end of this execution, the second track has at most $T(n)$ symbols, so erasing it takes time at most $3T(n)+6$.

Transferring all marks takes time at most $T(n)+2$.

Making the guesses of the transitions takes time at most $2T(n)+4$.

Finally, each simulation takes time $2T(n)+4$ (the extra factors account for the time we spend erasing the first tape by the end and rewinding the second tape).

Therefore M runs in time at most $O(T(n))$ as desired. $\triangleleft$

Exercise 2. Show that if $\mathrm{DSPACE}(n) \subseteq \mathrm{P}$, then $\mathrm{PSPACE} = \mathrm{P}$. Recall that $\mathrm{PSPACE} = \bigcup_{k\geqslant 1} \mathrm{DSPACE}(n^k)$. Hint: padding and PSPACE-completeness.

Solution. Recall first that the problem QBF of quantified boolean formula satisfiability is PSPACE-complete and $\mathrm{QBF} \in \mathrm{DSPACE}(n^2)$.

Recall also the definition of $\mathrm{pad}\colon \Sigma^* \times \mathbb{N} \to \Sigma^*$. We have

$$\mathrm{pad}(x,n) = x\$^i,$$

where $i = \max\{n - |x|, 0\}$.

Finally, recall that if $L \subseteq \Sigma^*$ and $f\colon \mathbb{N} \to \mathbb{N}$, then we have

$$\mathrm{pad}(L, f(n)) = \{\mathrm{pad}(x, f(|x|)) : x \in L\}.$$

Since n^2 is space constructible, the above readily implies that

$$\mathrm{pad}(\mathrm{QBF}, n^2) \in \mathrm{DSPACE}(n).$$

This is because checking the formatting $x\$^i$ in $\mathrm{pad}(\mathrm{QBF}, n^2)$ takes space at most n^2 and checking if $x \in L$ for a well-formatted $x\$^i$ string (i.e., such that $i = |x|^2 - |x|$) takes space $|x|^2$, which is the length of the input.

Note also that QBF is polynomially reducible to $\mathrm{pad}(\mathrm{QBF}, n^2)$ as we can simply compute $|x|^2$ and pad the input. This implies that $\mathrm{pad}(\mathrm{QBF}, n^2)$ is PSPACE-complete.

But then, since $\mathrm{pad}(\mathrm{QBF}, n^2) \in \mathrm{DSPACE}(n)$, it follows that if $\mathrm{DSPACE}(n) = \mathrm{P}$, then $\mathrm{PSPACE} = \mathrm{P}$ (since every PSPACE problem can be reduced to $\mathrm{pad}(\mathrm{QBF}, n^2)$ in polynomial time and $\mathrm{pad}(\mathrm{QBF}, n^2)$ would be solvable in polynomial time). $\triangleleft$

Exercise 3. Show that $\mathrm{DSPACE}(n) \neq \mathrm{P}$. Hint: reinspect Exercise 2 under the light of hierarchy theorems.

Solution. Suppose toward a contradiction that $\mathrm{DSPACE}(n) = \mathrm{P}$. By Exercise 2, we know that $\mathrm{PSPACE} = \mathrm{P}$. In particular, we have $\mathrm{DSPACE}(n^2) \subseteq \mathrm{PSPACE} = \mathrm{P} = \mathrm{DSPACE}(n)$.

But since $\lim_{n\to\infty} n/n^2 = 0$ and n^2 is space constructible (and $n \leqslant n^2$), by the Space Hierarchy Theorem, we know that $\mathrm{DSPACE}(n) \subsetneq \mathrm{DSPACE}(n^2)$, so the above is a contradiction. $\triangleleft$

Exercise 4. Show that $\mathrm{NSPACE}(n) \neq \mathrm{P}$.

Solution. Suppose toward a contradiction that $\mathrm{NSPACE}(n) = \mathrm{P}$. But then we have $\mathrm{DSPACE}(n) \subseteq \mathrm{NSPACE}(n) = \mathrm{P}$, which by Exercise 2 implies that $\mathrm{PSPACE} = \mathrm{P}$. In particular, we have $\mathrm{DSPACE}(n^3) \subseteq \mathrm{PSPACE} = \mathrm{P} = \mathrm{NSPACE}(n)$.

By Savitch's Theorem, we have $\mathrm{NSPACE}(n) \subseteq \mathrm{DSPACE}(n^2)$, so the above becomes $\mathrm{DSPACE}(n^3) \subseteq \mathrm{DSPACE}(n^2)$.

But since $\lim_{n\to\infty} n^2/n^3 = 0$ and n^3 is space constructible (and $n^2 \leqslant n^3$), by the Space Hierarchy Theorem, we know that $\mathrm{DSPACE}(n^2) \subsetneq \mathrm{DSPACE}(n^3)$, so the above is a contradiction. $\triangleleft$