CS 579. Computational Complexity
Problem Set 1 Solutions

Alexandra Kolla

February 7, 2017

In many solutions, we use the following definition: For any language A and function $f : \mathbb{N} \to \mathbb{N}$, let $A_{f(n)} = \left\{(x, 1^{f(|x|)}) \;\middle|\; x \in A\right\}$. Let $\text{pad}_{f(n)}(x) = (x, 1^{f(|x|)})$. Then

$$x \in A \iff \text{pad}_{f(n)}(x) \in A_{f(n)}$$

and $\text{pad}_{f(n)}$ is computable in polynomial time, so $A \leq_m^p A_{f(n)}$.

Problem 1

Recall that **NEXP** is defined by

$$\textbf{NEXP} = \bigcup_{c \geq 1} \textbf{NTIME}(2^{n^c}).$$

Give a definition of **NEXP** that does not involve non-deterministic Turing machines, analogous to the verifier definition of **NP** seen in class, and prove that your definition is equivalent to the above definition using non-deterministic Turing machines.

Solution to Problem 1

Define $\textbf{NEXP}_1 = \bigcup_{c \geq 1} \textbf{NTIME}\left(2^{n^c}\right)$.

Define the class $\textbf{NEXP}_2$ as follows: A language $L \in \textbf{NEXP}_2$ if there exists a relation $R(x, y)$ decidable in $p(|x|) \in O\left(2^{|x|^c}\right)$ for some $c \in \mathbb{N}_+$ such that

$$x \in L \iff \exists y, |y| \leq p(|x|) \wedge (x, y) \in R.$$

We'll show that $\textbf{NEXP}_1 = \textbf{NEXP}_2$.

Consider an arbitrary language $L \in \textbf{NEXP}_1$. Let M be a NDTM deciding L running in time $q(|x|) \in O\left(2^{|x|^c}\right)$ for some $c \in \mathbb{N}_+$. We'll define the relation $R(x, y)$ by

$$R = \{(x, y) \mid y \text{ is a sequence of non-deterministic choices for an accepting path of } M \text{ on input } x\}.$$

We observe that $R(x,y)$ is decidable in $q(|x|)$ time by exhibiting a DTM deciding $R(x,y)$ with the required running time. The desired TM N simply simulates M on x, deterministically choosing a single branch of M's computation at the ith step of M according to the ith bit of y. N runs in $\text{poly}(q(|x|)) \in O\left(2^{\text{poly}(|x|)}\right)$ time since any valid y has length $q(|x|)$ and looking up the ith bit of y between executing successive steps of M incurs only a polynomial slowdown over executing M. Furthermore, if $x \in L$ there is some accepting computation of M on input x, and if $x \notin L$ then all computations of M on input x are rejecting, so we obtain

$$x \in L \iff \exists y, |y| \leq q(|x|) \wedge (x,y) \in R.$$

We conclude that $L \in \mathbf{NEXP}_2$. Thus, $\mathbf{NEXP}_1 \subseteq \mathbf{NEXP}_2$.

Now consider an aritrary language $L \in \mathbf{NEXP}_2$. Let R be the relation required by the definition of $\mathbf{NEXP}_2$, and let M be a DTM deciding $R(x,y)$ in $p(|x|) \in O\left(2^{|x|^c}\right)$ time for some $c \geq 1$.

We'll construct a NDTM N deciding L as follows: Given input x, N will non-deterministically write down a string y of length $p(|x|)$ on the input tape following x, and then will simulate M on input (x,y), accepting x on a given path iff $M(x,y)$ accepts. By construction, N runs in $O(p(|x|))$ time, since simulating M takes $p(|x|)$ time and writing y takes $p(|x|)$ time. Furthermore $x \in L$ if and only if there is some y such that $M(x,y)$ accepts, so N will accept if and only if $x \in L$. Thus we have $L \in \mathbf{NEXP}_1$. We conclude that $\mathbf{NEXP}_2 \subseteq \mathbf{NEXP}_1$.

Finally, we conclude that $\mathbf{NEXP}_1 = \mathbf{NEXP}_2$.

Problem 2

Recall that $\mathbf{E} = \mathbf{DTIME}\left(2^{O(n)}\right)$ is the class of problems solvable by a deterministic Turing machine in time $2^{O(n)}$, where n is the length of the input. We say that a language A has a many-to-one polynomial time reduction to a language B, written $A \leq_m^p B$ if there is a polynomial time computable function $f(\cdot)$ such that for every instance x we have $x \in A \iff f(x) \in B$.

- Show that $\mathbf{NP}$ is closed under polynomial many-to-one reductions, that is $A \leq_m^p B$ and $B \in \mathbf{NP}$ implies $A \in \mathbf{NP}$.

- Show that if $\mathbf{E}$ were closed under many-to-one reductions, we would have a contradiction to the time hierarchy theorem. Conclude that $\mathbf{NP} \neq \mathbf{E}$.

Solution to Problem 2

1. Let A and B be languages with $B \in \mathbf{NP}$ and $A \leq_m^p B$. Then there exists a NDTM M that decides B in time $\text{poly}(|x|)$. By the definition of a many-to-one polynomial

reduction, there exists a polytime computable function $f(\cdot)$ such that $x \in A \iff f(x) \in B$. Consider the following algorithm for deciding A:

Compute $f(x)$ in polytime. Then run M on $f(x)$, accepting and rejecting according to $M(f(x))$. Trivially $M(f(x)) = 1 \iff x \in A$. Furthermore, $|f(x)| = O(\text{poly}(|x|))$ and $\text{poly}(\text{poly}(|x|)) = \text{poly}(|x|)$, so running M on $f(x)$ takes polynomial time. Thus we have a polynomial time algorithm on a NDTM to decide A and we conclude that $A \in \mathbf{NP}$.

2. Assume $\mathbf{E}$ is closed under polynomial many-to-one reductions. Then let $A \in \mathbf{EXP}$ be an arbitrary language in $\mathbf{EXP}$. Then there exists a TM M that decides A in time $O(2^{O(n^c)})$ for some $c \in \mathbb{N}$. Consider the padded language A_{n^2}. As stated above, $A \leq_m^p A_{n^2}$. Trivially, $A_{n^2} \in \mathbf{E}$ since we can decide $y = (x, 1^{|x|^c}) \in A_{n^2}$ in time $O(2^{|y|})$ by first checking if y is of the form $(x, 1^{|x|^c})$ in $\text{poly}(|y|)$ time, and then running M on x and returning the answer computed in time $O(2^{|y|})$. By assumption $\mathbf{E}$ is closed under polynomial many-to-one reductions, so $A \in \mathbf{E}$. Therefore $\mathbf{EXP} \subseteq \mathbf{E}$. Since $\mathbf{E} \subseteq \mathbf{EXP}$, we have $\mathbf{E} = \mathbf{EXP}$.

   But by the time-hierarchy theorem there exists a language L such that any algorithm deciding L must run in time greater than 2^{n^2} for infinitely many inputs, but L can be decided by an algorithm in $\mathbf{DTIME}(2^{O(n^2)})$ so $L \in \mathbf{EXP}$. Thus, $L \in \mathbf{EXP}$ but $L \notin \mathbf{E}$ so $\mathbf{E} \subsetneq \mathbf{EXP}$. This contradicts the conclusion that $\mathbf{E} = \mathbf{EXP}$ so we conclude our assumption that $\mathbf{E}$ was closed under polynomial many-to-one reductions was false.

   Finally, since $\mathbf{E}$ is not closed under polynomial many-to-one reductions and $\mathbf{NP}$ is closed under polynomial many-to-one reductions, it immediately follows that $\mathbf{NP} \neq \mathbf{E}$.

Problem 3

Show that $\mathbf{SPACE}(n) \neq \mathbf{NP}$.

Solution to Problem 3

Assume that $\mathbf{SPACE}(n) = \mathbf{NP}$. Then $\mathbf{SPACE}(n)$ must be closed under polynomial many-to-one reductions since $\mathbf{NP}$ is closed under polynomial many-to-one reductions as shown below. Consider an arbitrary language $A \in \mathbf{SPACE}(n^2)$. As shown above, $A \leq_m^p A_{n^2}$. Furthermore, $A_{n^2} \in \mathbf{SPACE}(n)$ since we can decide $y = (x, 1^{|x|^2}) \in A_{n^2}$ by running the algorithm for deciding A on x, and returning the answer computed. This will use $O(n)$ space since $|x|^2 = O(|y|) = O(n)$. Then $A \in \mathbf{SPACE}(n)$ since $\mathbf{SPACE}(n)$ is closed under polynomial many-to-one reductions by assumption. Thus, $\mathbf{SPACE}(n^2) \subseteq \mathbf{SPACE}(n)$ so $\mathbf{SPACE}(n) = \mathbf{SPACE}(n^2)$ since the other direction of containment trivially holds.

By the space-hierarchy theorem $\mathbf{SPACE}(n) \subsetneq \mathbf{SPACE}(n^2)$. This is a contradiction so our assumption that $\mathbf{SPACE}(n) = \mathbf{NP}$ must be false, and we conclude that $\mathbf{SPACE}(n) \neq \mathbf{NP}$.

Problem 4

Prove that if $\mathbf{P} = \mathbf{NP}$, then $\mathbf{EXP} = \mathbf{NEXP}$.

Solution to Problem 4

Assume that $\mathbf{P} = \mathbf{NP}$. Consider an arbitrary language $L \in \mathbf{NEXP}$. Then L is decided by a NDTM M running in time $t(n) \in O\left(2^{n^c}\right)$ for some $c \in \mathbb{N}$. We now observe that $L_{t(n)} \in \mathbf{NP}$ since we can decide $L_{t(n)}$ by verifying that the input given is of the form $(x, 1^{t(|x|)})$ and then running M on x. But then by assumption, $L_{t(n)} \in \mathbf{P}$. Let N be a DTM deciding $L_{t(n)}$ in time $q(n) \in \text{poly}(n)$. We can now decide L in deterministic exponential time as follows: First, replace x with $\text{pad}_{t(n)}(x)$ in $\text{poly}(t(|x|))$ time. Then run N on $\text{pad}_{t(n)}(x)$ in $q(O(t(|x|))) \in \text{poly}\left(2^{|x|^c}\right)$ time, accepting if and only if N accepts. This algorithm runs in $O\left(2^{n^d}\right)$ time for some $d \in \mathbb{N}$ and decides L, so we conclude that $L \in \mathbf{EXP}$. Thus, $\mathbf{NEXP} \subseteq \mathbf{EXP}$ and by definition we have $\mathbf{EXP} \subseteq \mathbf{NEXP}$, so we conclude that $\mathbf{EXP} = \mathbf{NEXP}$.

Problem 5

Suppose $L_1, L_2 \in \mathbf{NP} \cap \mathbf{coNP}$. Show that the language

$$L_1 \triangle L_2 = \{x \mid x \text{ is in exactly one of } L_1, L_2\}$$

is in $\mathbf{NP} \cap \mathbf{coNP}$.

Solution to Problem 5

Let $L = A \triangle B$ for some $A, B \in \mathbf{NP} \cap \mathbf{coNP}$. We will show that $L \in \mathbf{NP} \cap \mathbf{coNP}$. Let M_1 and M_0 be polynomial-time NDTMs deciding A and $\bar{A}$ respectively, and let N_1 and N_0 be NDTMs deciding B and $\bar{B}$ respectively.

To show that $L \in \mathbf{NP}$ we exhibit an $\mathbf{NP}$ algorithm to decide L. Non-deterministically guess a bit i, and then simulate $M_i(x)$ and $N_{1-i}(x)$, and accept if both M and N both accept. Now $x \in L$ if and only if either $x \in A \cap \bar{B}$ or $x \in \bar{A} \cap B$ if and only if there is a choice of i such that $M_i(x)$ and $N_{i-1}(x)$ both accept.

To show that $L \in \mathbf{coNP}$, we'll show that $\bar{L} \in \mathbf{NP}$ by exhibiting an almost identical algorithm, modified to simulate $M_i(x)$ and $N_i(x)$ for a given choice of i. Correctness follows by an analogous argument.