CS 535: Complexity Theory, Fall 2023

Homework 6

Due: 11:59PM, Tuesday, October 31, 2023.

Reminder. Homework must be typeset with LaTeX preferred. Make sure you understand the course collaboration and honesty policy before beginning this assignment. Collaboration is permitted, but you must write the solutions *by yourself without assistance*. You must also identify your collaborators. Assignments missing a collaboration statement will not be accepted. Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.

Problem 0 (Term Paper Topic)**.** Your term paper topic and partner (if applicable) are due on Gradescope at the same time this homework assignment is. Instructions for the term paper are here: `https://cs-people.bu.edu/mbun/courses/535_F23/handouts/term_paper.pdf` and a list of suggested topics is here: `https://piazza.com/class/lm5tyuo0a7j2fu/post/89`.

Problem 1 ($\mathbf{NC}^1$ Captures Poly-Size Boolean Formulas)**.** For this problem, all circuits have fan-in 2 $\{\wedge, \vee\}$ gates, and you can assume all negations are pushed to the bottom. A Boolean formula is a special case of a circuit where every intermediate $\wedge$ or $\vee$ gate has fan-out 1. To make the accounting in this problem easier, we'll write "C has s gates" to mean that C has s $\{\wedge, \vee\}$ gates, explicitly <u>not</u> including the $2n$ inputs and their negations.

(a) Suppose $f : \{0,1\}^n \to \{0,1\}$ is computed by a <u>circuit</u> with s gates and depth d. Show that f is also computed by a <u>formula</u> of size $O(2^d \cdot s)$. (2 points)

Hint: Induction on the depth.

(b) Now we'll get to work on the converse, showing how to convert a formula into a low-depth circuit. First, prove the following fact about binary trees: For every binary tree with s vertices, there exists a vertex v such that the subtree rooted at v contains between $s/3$ and $2s/3$ vertices. (2 points)

(c) Use part (b) to show that every formula F with s gates has an equivalent formula $F' = (A \wedge B) \vee (C \wedge D)$, where each subformula A, B, C, D has at most $2s/3$ gates. (2 points)

(d) Use part (c) recursively to show that every formula with s gates has an equivalent formula of depth $O(\log s)$ and $\mathrm{poly}(s)$ gates. (2 points)

(e) Combine parts (a) and (d) to conclude that a language is in the complexity class $\mathbf{NC}^1$ if and only if it is also computable by a poly-size family of formulas. (2 points)

Problem 2 (More Time-Space Tradeoffs)**.** In class (and in Arora-Barak) we saw that $\mathbf{NTIME}(n) \not\subseteq \mathbf{TISP}(n^{1.2}, n^{0.2})$, and hence SAT cannot be solved by a deterministic TM running in, say, time $O(n^{1.1})$ and space $O(n^{0.1})$ simultaneously. In this problem, you'll similarly prove that co-nondeterministic linear time cannot be simulated in small *nondeterministic* time and space, and how far you can push the technique to get different tradeoffs. Assume every function you encounter in this problem is time- and space-constructible.

For time bound $T(n)$ and space bound $S(n)$, define $\mathbf{NTISP}(T, S)$ to be the class consisting of languages L such that L is decidable by a nondeterministic TM running in both time $O(T(n))$ and space $O(S(n))$. Define $\mathbf{coNTIME}(T(n)) = \mathbf{\Pi}_1\mathbf{TIME}(T(n))$ to be the class of languages decidable by a alternating TM of $\mathbf{\Pi}_1$ type in time $O(T(n))$.

(a) Generalize Claim 5.11.1 in Arora-Barak to prove that for $T(n) \geq n^2$ and $S(n) \geq \log n$, we have $\mathbf{NTISP}(T, S) \subseteq \mathbf{\Sigma}_2\mathbf{TIME}(\sqrt{TS})$. (3 points)

(b) Generalize Claim 5.11.2 in Arora-Barak to prove that if $\mathbf{coNTIME}(n) \subseteq \mathbf{NTIME}(n^c)$ for some $c > 1$, then $\mathbf{\Sigma}_2\mathbf{TIME}(f(n)) \subseteq \mathbf{NTIME}((f(n))^c)$. (3 points)

(c) First we'll see how large we can make the time requirement. Use parts (a) and (b), together with the following Fact (which you can assume without proof), to prove that for every $c < \sqrt{2}$, there exists a $\delta > 0$ such that $\mathbf{coNTIME}(n) \not\subseteq \mathbf{NTISP}(n^c, n^\delta)$. (2 points)

Fact 1. *For all positive constants $b > a > 0$, we have*

$$\mathbf{coNTIME}(n^b) \not\subseteq \mathbf{NTIME}(n^a).$$

Hint: Note that δ is allowed to depend on c. You'll want to choose δ small enough so that $c(c + \delta) < 2$.

You don't have to show it, but this implies that TAUT cannot be solved by a nondeterministic TM using $O(n^{1.41\cdots})$ time and $n^{o(1)}$ space.

(d) Now we'll see how far we can push the space requirement. Prove that for every $c < 1$, there exists a $\delta > 0$ such that $\mathbf{coNTIME}(n) \not\subseteq \mathbf{NTISP}(n^{1+\delta}, n^c)$. This result implies that TAUT cannot be solved by a nondeterministic algorithm using $n^{1+o(1)}$ time and $O(n^{0.999})$ space. Hint: This time, choose δ small enough so that $(c+1+\delta)(1+\delta) < 2$. (2 points)

(e) (*Bonus*) Prove Fact 1.

Problem 3 (*Bonus* Improved Deterministic Time-Space Tradeoffs)**.** Let's go back to the setting of deterministic time-space tradeoffs for $\mathbf{NTIME}(n)$. In this problem, you'll see how to get even better tradeoffs by repeatedly trading alternations for time.

(a) Suppose $\mathbf{NTIME}(n) \subseteq \mathbf{DTIME}(n^c)$ for some $c > 1$. Show that $\mathbf{TISP}(T, S) \subseteq \mathbf{coNTIME}((TS^2)^{c^2/(2+c)})$. Use this to conclude that $\mathbf{NTIME}(n) \not\subseteq \mathbf{TISP}(n^c, n^{o(1)})$ whenever $c^3 < 2 + c$, i.e., $c < 1.521\ldots$. Hint: Let C_0, C_f be the start and accept configurations of a deterministic TM running in time T. Then C_f is reachable from C_0 in T time steps iff *for all* $C' \neq C_f$, we have that C' is *not* reachable from C_0 in T time steps.

(b) Generalize the above argument inductively to show that $\mathbf{NTIME}(n) \not\subseteq \mathbf{TISP}(n^c, n^{o(1)})$ whenever $c(c-1) < 1$, i.e., $c < \phi = 1.618\ldots$.