KØBENHAVNS UNIVERSITET

Computability and Complexity: Problem Set 2

Due: Tuesday March 7 at 23:59 AoE .
Submission: Please submit your solutions via *Absalon* as a PDF file. State your name and e-mail address close to the top of the first page. Solutions should be written in LaTeX or some other math-aware typesetting system with reasonable margins on all sides (at least 2.5 cm). Please try to be precise and to the point in your solutions and refrain from vague statements. Make sure to explain your reasoning. *Write so that a fellow student of yours can read, understand, and verify your solutions.* In addition to what is stated below, the general rules for problem sets stated on the course webpage always apply.
Collaboration: Discussions of ideas in groups of two to three people are allowed—and indeed, encouraged—but you should always write up your solutions completely on your own, from start to finish, and you should understand all aspects of them fully. It is not allowed to compose draft solutions together and then continue editing individually, or to share any text, formulas, or pseudocode. Also, no such material may be downloaded from or generated via the internet to be used in draft or final solutions. Submitted solutions will be checked for plagiarism. You should also clearly acknowledge any collaboration. State close to the top of the first page of your problem set solutions if you have been collaborating with someone and if so with whom. *Note that collaboration is on a per problem set basis, so you should not discuss different problems on the same problem set with different people.*
Reference material: Some of the problems are "classic" and hence it might be possible to find solutions on the Internet, in textbooks or in research papers. It is not allowed to use such material in any way unless explicitly stated otherwise. Anything said during the lectures or in the lecture notes should be fair game, though, unless you are specifically asked to show something that we claimed without proof in class. All definitions should be as given in class or in Arora-Barak and cannot be substituted by versions from other sources. It is hard to pin down 100% watertight, formal rules on what all of this means—when in doubt, ask the main instructor.
Grading: A score of 100 points is guaranteed to be enough to pass this problem set.
Questions: Please do not hesitate to ask the instructor if any problem statement is unclear, but please make sure to send private messages sometimes specific enough questions could give away the solution to your fellow students, and we want all of you to benefit from working on, and learning from, the problems. Good luck!

1 (20 p) We proved in class that there are oracles relative to which P and NP are equal by defining the language $\textsc{ExpCom} = \{\langle M, x, 1^n\rangle \mid M \text{ accepts } x \text{ within } 2^n \text{ steps}\}$ and showing that $\mathsf{P}^{\textsc{ExpCom}} = \mathsf{NP}^{\textsc{ExpCom}} = \mathsf{EXP}$. In this problem we want to understand how important (or unimportant) the exact details in the definition of $\textsc{ExpCom}$ is for this result to hold.

1a Let $\textsc{ExpCom}' = \{\langle M, x, 1^n\rangle \mid M \text{ accepts } x \text{ within } n \text{ steps}\}$. Do the equalities $\mathsf{P}^{\textsc{ExpCom}'} = \mathsf{NP}^{\textsc{ExpCom}'} = \mathsf{EXP}$ hold? Modify the argument we gave in class to establish these equalities or explain where the proof fails.

1b Let $\textsc{ExpCom}'' = \{\langle M, x, n\rangle \mid M \text{ accepts } x \text{ within } 2^n \text{ steps}\}$ (where n in the input is a number given in binary). Does it hold that $\mathsf{P}^{\textsc{ExpCom}''} = \mathsf{NP}^{\textsc{ExpCom}''} = \mathsf{EXP}$? Adapt the proof given in class or explain where it fails.

2 (10 p) We say that a language $L \subseteq \{0,1\}^*$ is *sparse* if there is a polynomial p such that it holds for every $n \in \mathbb{N}^+$ that $|L \cap \{0,1\}^n| \leq p(n)$. Show that if L is sparse, then $L \in \mathsf{P/poly}$.

3 (30 p) Consider the language

$$\textsc{SpaceBoundedTM} = \{\langle M, x, 1^n\rangle \mid M \text{ accepts } x \text{ in space } n\}$$

where M is a deterministic Turing machine and 1^n denotes a string of ones of length n (as usual). Prove that SpaceBoundedTM is PSPACE-complete from first principles (i.e., prove that SpaceBoundedTM is in PSPACE and that any other language in PSPACE reduces to it).

(In this problem, we assume that all Turing machines have a fixed configuration in terms of alphabet and number of tapes, and that a universal TM for space-bounded computation as in Exercise 4.1 in Arora-Barak can be assumed without proof.)

4 (30 p) Let us say that a function $f : \{0,1\}^* \to \{0,1\}^*$ is *write-once logspace computable* if f can be computed by a Turing machine M that uses $\mathrm{O}(\log n)$ space on its work tapes and whose output tape is *write-once*. By a write-once tape we mean a tape where at every time step M can either keep its head at the same position on the tape or write a symbol to it and move one location to the right, but M can never read from the tape or move left. The used cells on the write-once tape are not counted towards the space bound on M.

Prove that f is write-once logspace computable if and only if it is *implicitly logspace computable* as defined in class.

5 (40 p) Recall that we write $\Pi_i^p = \mathsf{co}\Sigma_i^p$ to denote the complement of the complexity class Σ_i^p and that the union of all such classes form the polynomial hierarchy $\mathsf{PH} = \bigcup_{i\in\mathbb{N}^+} \Sigma_i^p = \bigcup_{i\in\mathbb{N}^+} \Pi_i^p$. If it would hold that $\mathsf{PH} = \Sigma_i^p$ one says that "the polynomial hierarchy collapses to the ith level" (which, as we said in class, is generally not believed to be the case).

Prove that if $\Sigma_i^p = \Pi_i^p$, then the polynomial hierarchy collapses to the ith level.

6 (40 p) In our lectures on Boolean circuits we defined $\mathsf{DTIME}\big(T(n)\big)/a(n)$ as the class of languages decided by Turing machines M running in time $\mathrm{O}\big(T(n)\big)$ that also get a specific advice string $\alpha_n \in \{0,1\}^{a(n)}$ for inputs of size n. We then proved (or at least outlined a proof) that $\mathsf{P/poly} = \bigcup_{c,d\in\mathbb{N}^+} \mathsf{DTIME}\big(n^c\big)/n^d$.

Is it possible to change the definition so that not only the advice string α_n depends on the size of the input, but so that we can also pick different Turing machines M_n for different input sizes (while still maintaining that all running times be bounded by a common polynomial $p(n)$), and prove that $\mathsf{P/poly}$ is equal to the set of languages decided by such sequences of Turing machines $\{M_n\}_{n\in\mathbb{N}^+}$ with advice strings $\{\alpha_n\}_{n\in\mathbb{N}^+}$? Work out the details to show that this alternative definition is just as fine, or give a clear mathematical argument why it seems problematic.

7 (60 p) Show that $\mathsf{P} \neq \mathsf{SPACE}\big(n^k\big)$ for any fixed $k \in \mathbb{N}^+$.
Hint: Use padding. (Also, just to avoid confusion, note that $\mathsf{P} \subseteq \bigcup_{k\in\mathbb{N}^+} \mathsf{SPACE}\big(n^k\big) = \mathsf{PSPACE}$, and of course we do not know that $\mathsf{P} \neq \mathsf{PSPACE}$, but the point here is that we are fixing k.)