KØBENHAVNS UNIVERSITET

Computability and Complexity: Problem Set 4

Due: Monday April 3 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!

PLEASE NOTE that this is still a draft version (last updated March 19, 2023). There might be one or (at most) two problems added at a later date. However, the problems listed below will appear also in the final version, so you can start working on them already now.

1 (10 p) Prove that any monotone Boolean function can be computed by a monotone Boolean circuit.

2 (20 p) When we used the monotone circuit lower bound for clique to obtain an exponential lower bound on resolution refutations of clique-colouring formulas, the form of the circuit lower bound that we used was that small monotone circuits cannot distinguish between m-cliques and $(m-1)$-colourable graphs (see Theorem 5.1 in the LaTeX:ed lecture notes).

Suppose that we would instead have been given the circuit lower bound stated in Theorem 4 in the notes for Lecture 16 without knowing anything about how this lower bound had been established. Could we still obtain the clique-colouring formula lower bound shown in class, or would the argument fail? Please indicate how to adapt the proof or explain where it breaks.

3 (30 p) For a language $L \subseteq \{0,1\}^*$, let $L_k = \{x \in L; |x| \leq k\}$ denote all strings in L of length at most k. We say that L is *downward self-reducible* if there is a polynomial-time algorithm A that given x and oracle access to $L_{|x|-1}$ decides correctly whether $x \in L$ or not.

Prove that if a language L is downward self-reducible, then it must hold that $L \in \mathsf{PSPACE}$.

4 (30 p) In our lectures on proof complexity, we defined a *resolution refutation* $\pi : F \vdash \bot$ of an unsatisfiable CNF formula F to be a sequence of clauses $\pi = (C_1, C_2, \ldots, C_L)$ such that each C_i is either a clause in F (an *axiom*) or is derived from two clauses $C_j, C_k \in \pi$, $j < k < i$, by the *resolution rule*

$$\frac{C \vee x \qquad D \vee \overline{x}}{C \vee D},$$

and such that the final clause C_L is the empty clause containing no literals, denoted $\bot$. The *length* of the refutation π is L.

When proving results about resolution, it is often convenient to also allow a clause $C_i \in \pi$ to be derived from some C_j, $j < i$, by the *relaxation rule*

$$\frac{C}{C \vee D},$$

where one deduces the strictly weaker clause $C \vee D$ from C. Show that adding this rule does not really change the proof system. Formally, prove that if $\pi : F \vdash \bot$ is a resolution refutation of an unsatisfiable CNF formula F using also the relaxation rule, then there is a standard resolution refutation $\pi' : F \vdash \bot$ in at most the same length without any applications of relaxation.
Hint: Use induction over the sequence of clauses $\pi = (C_1, C_2, \ldots, C_L)$ in the relaxed resolution refutation.

5 (40 p) In the monotone circuit lower bound for clique presented in Lectures 16 and 17, we used the concept of *sunflowers*, which are collections of sets $X_1, \ldots, X_p$ such that there is a set X with the property that $X_i \cap X_j = X$ for all $1 \leq i < j \leq p$. This is quite a strict requirement, and so one can ask whether the definition could be relaxed a bit without breaking the lower bound proof (and perhaps even yielding a slightly better bound).

5a Let us say that a collection of sets $X_1, \ldots, X_p$ is a *sub-sunflower* if there is a set X with the property that $X_i \cap X_j \subseteq X$ for all $1 \leq i < j \leq p$. Would the lower bound we did in class still go through if we did the "plucking" with sub-sunflowers? Please explain how to adapt the argument or point out where it fails.

5b Say that $X_1, \ldots, X_p$ form a *super-sunflower* if there is a set X such that $X_i \cap X_j \supseteq X$ for all $1 \leq i < j \leq p$. Would the lower bound we did in class still go through if we used super-sunflowers? Please explain how to adapt the argument or point out where it fails.

A general comment is that you should not expect to have to write pages of detailed arguments in the cases where you believe the proof still works or can be adapted to work. Also, when it seems that the proof cannot be made to work you do not have to prove beyond all doubt that no way of formalizing an argument along similar lines can possibly work in any universe—it is enough to point out, briefly but concretely, what technical difficulties arise, and why they seem hard to circumvent.

6 (60 p) Suppose that $A(\mathbf{p}, \mathbf{q}) \wedge B(\mathbf{p}, \mathbf{r})$ is an unsatisfiable CNF formula over disjoint sets of variables $\mathbf{p}, \mathbf{q}, \mathbf{r}$. In this problem we want to go over the proof of Theorem 5.3 in the LaTeX:ed lecture notes and fill in some of the missing details.

6a Describe the details of how the clause is constructed in case 3 (the "$\mathbf{r}$-case") in part 2 of Theorem 5.3 and prove that the inductive hypotheses are maintained.

6b Describe the details of how the circuit gate is constructed in case 3 in part 1 of Theorem 5.3 and prove that this gate computes the type of the clause correctly.

6c Under the assumption that $\mathbf{p}$-variables appear only negatively in $B(\mathbf{p}, \mathbf{r})$, show that if $A(\mathbf{p}, \mathbf{q}) \wedge B(\mathbf{p}, \mathbf{r})$ has a resolution refutation of length L, then there exists a monotone Boolean interpolating circuit $I(\mathbf{p})$ of size $O(L)$.

Remark: Note that there is no need to reproduce the proofs covered in class—you can assume that they are all known. Instead, just explain in a precise manner how to fill in the missing details, and then motivate (potentially briefly, but clearly and to the point) why the proof works. Also note that the subproblems above can be solved independently of each other.