KØBENHAVNS UNIVERSITET

Computability and Complexity: Problem Set 3

Due: Thursday March 19 at 23:59 AoE.
Submission: Please submit your solutions via *Absalon* as a PDF file. State your name and e-mail address at 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. Never just state an an answer, but make sure to also explain your reasoning. *Write so that a fellow student of yours can read, understand, and verify your solutions.* In addition to what is said 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, or any material found in Arora–Barak, 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 total score of 70 points will be enough for grade 02, 100 points for grade 4, 130 points for grade 7, 160 points for grade 10, and 200 points for grade 12 on this problem set. Please note that problems are not necessarily presented in order of difficulty. Unless otherwise stated, every subproblem can be solved independently of other subproblems. Any revised versions of the problem set with clarifications and/or corrections will be posted on the course webpage `jakobnordstrom.se/teaching/CoCo26/`.
Questions: Please do not hesitate to ask the instructors or TA if any problem statement is unclear, but please make sure to send private messages when using Absalon—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) Show that $\mathsf{ZPP} = \mathsf{RP} \cap \mathsf{coRP}$.

2 (20 p) This problem demonstrates that "circuits do not really need randomness", or, in other words, that randomness can be replaced by non-uniformity.

Let C be a Boolean circuit of size $s \geq n$ over the variables $x_1, \ldots, x_n$ and $r_1, \ldots, r_m$. Suppose that $f : \{0,1\}^n \to \{0,1\}$ is a Boolean function such that for all $x \in \{0,1\}^n$ it holds that if $r \in \{0,1\}^m$ is chosen uniformly at random then

$$\Pr_r[C(x,r) = f(x)] \geq \frac{2}{3} .$$

Prove that there is a Boolean circuit D of size at most $\text{poly}(s)$ over the variables $x_1, \ldots, x_n$ such that for all $x \in \{0,1\}^n$ it holds that $D(x) = f(x)$.

3 (40 p) This problem is about balancing Boolean formulas (which are Boolean circuits when the underlying graph is a rooted directed tree). Let $f : \{0,1\}^n \to \{0,1\}$ be computed by a Boolean formula of size s. Show that f can also be computed by a Boolean formula of size at most $\text{poly}(s)$ and depth at most $\text{O}(\log s)$.

4 (30 p) This problem is about transitivity of log-space reduction. Recall from class that $\leq_\ell$ denotes log-space reducibility (defined via implicitly log-space computable reductions). Let $A, B, C \subset \Sigma^*$ be languages. Show that if $A \leq_\ell B$ and $B \leq_\ell C$, then $A \leq_\ell C$.

5 (40 p) A *decision tree* T is a binary tree with edges directed from the root to the leaves and with leaves labelled 0/1, non-leaves labelled by variables x_i, and the two edges out of every non-leaf labelled 0 and 1, respectively. We say that T *represents* the Boolean function $f : \{0,1\}^n \to \{0,1\}$ if for every assignment α it holds that when starting in the root of T and following the edge labelled by $\alpha(x_i)$ out of every non-leaf labelled by x_i we end up in a leaf labelled by the value $f(\alpha)$. The *depth* of a decision tree T is the length of a longest root-to-leaf path in T.

In this problem we want to study some connections between decision trees, CNF formulas, and DNF formulas.

5a Suppose that a Boolean function f can be represented as a decision tree of depth d. Show that f can also be represented as a d-CNF formula and as a d-DNF formula.

5b Suppose that a Boolean function f can be written both as a k-CNF formula and as an ℓ-DNF formula. Show that this implies that f also can be represented as a decision tree of depth at most $k\ell$.

6 (40 p) In this problem, you will show that any Boolean function $f : \{0,1\}^n \to \{0,1\}$ has a circuit of size $O(2^n/n)$, meaning that Shannon's lower bound is tight up to constant factors.

6a Assume $\ell \leq n$. Show that there is a *multi-output* circuit C_ℓ (i.e., a Boolean circuit with many output gates) of size $O\big(2^{2^\ell}\big)$ that computes all Boolean functions on ℓ bits.

6b Define the function $A_m : \{0,1\}^{m+2^m} \to \{0,1\}$ as follows. We call the first m input bits $u_1, \ldots, u_m$ and the remaining $v_0, \ldots, v_{2^m-1}$. On a given input (u, v), the function interprets x as the binary encoding of a number k in the range $\{0, \ldots, 2^m - 1\}$ and outputs v_k. Show that A_m has a Boolean circuit D_m of size $O(2^m)$.

6c Using $D_{n-\ell}$ and C_ℓ, show that for any $\ell \leq n$ we can construct a circuit for $f(x_1, \ldots, x_n)$ of size $O(2^{n-\ell} + 2^{2^\ell})$.

Hint: Use C_ℓ to compute all functions of the last ℓ bits $x_{n-\ell+1}, \ldots, x_n$.

6d Show that we can choose a suitable value for the parameter ℓ to get a circuit of size $O(2^n/n)$.

7 (50 p) Let multiprover interactive protocols be defined as the interactive protocols in Section 8.1 in Arora-Barak, except that there are several provers and that the verifier's messages in each round depends on previous messages from all provers (and on the verifier's private randomness). The messages sent by each prover only depends on the communication with the verifier, however, just as before. Let $\mathsf{MIP}[N]$ denote the set of languages that can be decided by N-multiprover interactive protocols in a polynomial number of rounds (in analogy with $\mathsf{IP} = \mathsf{MIP}[1]$ in Definition 8.6 in Arora-Barak).

Prove that, as claimed in class, only two provers are needed to realize the full power of multiprover interactive protocols. That is, prove that $\mathsf{MIP}[2] = \mathsf{MIP}[\mathrm{poly}]$, where $\mathsf{MIP}[\mathrm{poly}]$-protocols have a number of provers scaling polynomially with the size of the input.

8 (70 p) Recall that we learned in class that counting the number of satisfying assignments for a CNF formula or counting the number of perfect matchings in a bipartite graph are (probably) both hard tasks. In this problem, we are concerned with the problem of counting the number of spanning trees of a graph.

Let $G = ([n], E)$ be a simple undirected graph. Denote by $st(G)$ the number of spanning trees of G. For an edge $e \in E$, denote by $G - e$ the graph obtained from G by deleting e, and denote by $G|e$ the graph obtained from G by contracting e (so that the two endpoints of $e = (u, v)$ are merged into a single vertex, which is connected by edges to the all the neighbours of u or v).

Denote by $L = L_G$ the $n \times n$ *Laplacian matrix* of G, i.e., the matrix such that the diagonal entries $L_{i,i}$ are the degrees of the vertices i and for off-diagonal entries of the matrix it holds that $L_{i,j} = L_{j,i} = -1$ if (i, j) is an edge and all the other entries are zero.

For an $n \times n$ matrix M, we let $det(M)$ denote the determinant of M. We write $M^{\ominus}[i]$ to denote the $(n-1) \times (n-1)$ matrix obtained by deleting the ith row and column from M. Similarly, we let $M^{\ominus}[i, j]$ denote the $(n-2) \times (n-2)$ matrix obtained by deleting the ith and jth rows and columns from M.

8a Prove that if $e \in E$ is an edge in G, then it holds that

$$st(G) = st(G - e) + st(G|e) \ .$$

8b Prove that if $e = (i, j) \in E$ is an edge in G, then it holds that

$$det(L_G^{\ominus}[i]) = det(L_{G-e}^{\ominus}[i]) + det(L_G^{\ominus}[i, j])$$

8c Prove that if $e = (i, j) \in E$ is an edge in G, then it holds that

$$det(L_G^{\ominus}[i, j]) = det(L_{G|e}^{\ominus}[j])$$

8d Describe a polynomial time algorithm for computing $st(G)$.

Remark: To get partial credit, you can use the results claimed in some of the subproblems to solve other subproblems.

9 (60 p) The goal of this exercise is to give a complete proof that PSPACE $\subseteq$ IP, strengthening the result coNP $\subseteq$ IP that was proven in class.

Given a quantified Boolean formula (QBF) $\psi = \forall x_1 \exists x_2 \forall x_3 \cdots \exists x_n \, \phi(x_1, \ldots, x_n)$, we can use arithmetization as in our proof of coNP $\subseteq$ IP to construct a polynomial P_ϕ such that ψ is true if and only if $\prod_{b_1 \in \{0,1\}} \sum_{b_2 \in \{0,1\}} \prod_{b_3 \in \{0,1\}} \cdots \sum_{b_n \in \{0,1\}} P_\phi(b_1, \ldots, b_n) \neq 0$. However, the SumCheck protocol we used to decide the $\#\text{Sat}_D$ problem for CNF formulas no longer works, since each multiplication corresponding to a $\forall$-quantifier can double the degree of the polynomial.

9a (20 p) Suppose that ψ is a QBF formula (not necessarily in *prenex normal form* as described in Definition 4.10 and discussed further below in Arora-Barak) satisfying the following property: if $x_1, \ldots, x_n$ are the variables of ψ sorted in order of first appearance, then for every variable x_i there is at most a single universal quantifier involving x_j for any $j > i$ appearing before the last occurrence of x_i in ψ. Show that in this case, when we run the SumCheck protocol with the modification that we check $s(0) \cdot s(1) = K$ for product operations (i.e., $\forall$-quantifiers), the prover only needs to send polynomials of degree $O(n)$ since the degree blow-up is at most a constant factor 2.

9b (20 p) Assuming that any QBF formula ψ can be rewritten to satisfy the property in Problem 9a, use this to show that TQBF $\in$ IP and hence PSPACE $\subseteq$ IP.

9c (20 p) Show that any QBF formula ψ of size m can be transformed into a logically equivalent formula ψ' of size $O(m^2)$ that satisfies the property in Problem 9a.

Hint: Introduce a new variable y_i for any occurrence of x_i that we need to get rid of and encode that x_i and y_i take the same truth value.