KTH Computer Science and Communication

Computational Complexity: Problem Set 3

Due: Tuesday November 24, 2015, at 23:59 AoE. Submit your solutions as a PDF file by e-mail to `jakobn at kth dot se` with the subject line `Problem set 3: ⟨your full name⟩`. Name the PDF file `PS3_⟨YourFullName⟩.pdf` with your name written in CamelCase without blanks and in ASCII without national characters. State your name and e-mail address at the very top of the first page. Solutions should be written in LaTeX or some other math-aware typesetting system with reasonable margins on all sides (ca 2.5 cm). Please try to be precise and to the point in your solutions and refrain from vague statements. *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 stated on the course webpage always apply.

Collaboration: Discussions of ideas in groups of two people are allowed—and indeed, encouraged—but you should write down your own solutions individually and understand all aspects of them fully. You should also acknowledge any collaboration. State at the very 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 easy 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 which can be found in chapters of Arora-Barak covered in the course, 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 definitions from other sources. It is hard to pin down 100% watertight formal rules on what all of this means—when in doubt, ask the lecturer.

About the problems: Some of the problems are meant to be quite challenging and you are not necessarily expected to solve all of them. A total score of around 85 points should be enough for grade E, 125 points for grade D, 165 points for grade C, 205 points for grade B, and 245 points for grade A on this problem set. Any corrections or clarifications will be given at `piazza.com/kth.se/fall2015/dd2445/` and any revised versions will be posted on the course webpage `www.csc.kth.se/DD2445/kplx15/`.

1 (10 p) Show that if one-way functions exist, then $\mathsf{P} \neq \mathsf{NP}$.

2 (20 p) Show that $\mathsf{BP} \cdot \mathsf{NP} = \mathsf{AM}[2]$.

3 (30 p) Recall that we defined an *encryption scheme* for plaintexts $x \in \{0,1\}^m$ with encryption keys $k \in \{0,1\}^n$ to be a pair of functions $\big(E(k,x), D(k,x)\big) = \big(E_k(x), D_k(x)\big)$ such that for every key k and plaintext x it holds that $D_k(E_k(x)) = x$. An encryption scheme is *perfectly secret* if for every pair of plaintext messages $x, x' \in \{0,1\}^m$ it holds that the distributions $E_{U_n}(x)$ and $E_{U_n}(x')$ are identical (where U_n denotes the uniform distribution over $\{0,1\}^n$).

We claimed in class that no encryption scheme (E, D) with $m > n$ can be perfectly secure. Prove that this is so.

Hint: What happens if all distributions $E_{U_n}(x)$ have the same support?

4 (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 $L \in \mathsf{PSPACE}$.

5 (30 p) Consider the following proposed interactive zero knowledge protocol for graph isomorphism:

Input: G_0, G_1 and private permutation π for Prover such that $G_1 = \pi(G_0)$.

Verifier: Sends random permutation π^* and random bit b_v.

Prover: Sends random bit b_p.

Verifier: Sends graph $H = \pi^*(G_{b_p})$.

Prover: Sends permutation π_H.

Verifier: Accepts if and only if $\pi_H(H) = G_{b_v}$.

Analyze whether this protocol is complete, sound, and/or zero knowledge. For a full score you need to provide formal proofs for each of the three properties establishing that the property in question holds or fails to hold.
Remark: In order for this problem to make sense, we are tacitly making the assumption that the graph isomorphism problem in itself is not efficiently decidable in randomized polynomial time. (This might well be false, of course, but that is a bit beside the point—the real focus of this problem is on zero knowledge protocols, not graph isomorphism.)

6 (40 p) In our lectures on proof complexity, we defined the CNF encoding of the (negation of the) pigeonhole principle PHP_n^m for any number of pigeons m and pigeonholes n, but then focused on $m = n + 1$ when proving the $\exp(\Omega(n))$ lower bound on resolution refutation length for the formulas PHP_n^{n+1}.

What would happen with this lower bound proof if we considered more than $n + 1$ pigeons, say $m = n + 2$, $m = 2n$, $m = n^2$, or even $m = 2^n$ pigeons? Would the proof still work, and would we still get a lower bound on the form $\exp(\Omega(n))$? Describe how to adapt the proof to work for larger m; determine for how large m you can make it work; and/or explain when or why the approach we used in class fails.
Remark: In order to solve this problem, it is not necessary to give a full answer to the question of how the hardness of the formula PHP_n^m depends on m—it is fully sufficient to analyze the concrete lower bound approach that we employed in class and try to understand how far this technique can (or cannot) be pushed. You do not need to prove all claims you make beyond reasonable doubt—in particular, it is not necessary to prove any claims that we left unproven in class—but it should be possible to see how to plausibly fill in any gaps in your arguments.

7 (40 p) When proving a lower bound on resolution refutation length, we studied a prosecutor-defendant game and proved a lower bound on the size of a prosecutor strategy for PHP_n^{n+1} in this game. It is not hard to see that the same game can be played on any unsatisfiable CNF formula F (which the defendant claims to be satisfiable), where the prosecutor asks about assignments to variables $x \in \mathit{Vars}(F)$, or forgets such assignments, and the "explicit contradictions" the prosecutor is trying to force are partial assignments falsifying some axiom clause $C \in F$. The same reasoning we used in class shows that any resolution refutation of F in length L yields a strategy for the prosecutor of size $\mathrm{O}(L)$ (i.e., with $\mathrm{O}(L)$ rules in the instruction book).

In this problem we are interested in the other direction. Suppose that the prosecutor has a strategy for some formula F that requires consideration only of L cases in order to secure the conviction of the defendant. Can such a strategy be converted to a resolution refutation of F in length $\mathrm{O}(L)$? Describe how to convert a prosecutor strategy to a resolution refutation in essentially the same size, or explain why it seems hard to do the transformation in this other direction.

8 (50 p) In this problem we want to work out some of the missing details in our discussion in class concerning that cutting planes is exponentially stronger than resolution.

8a (20 p) Prove that if a CNF formula F is refutable in resolution in length L, then cutting planes can refute F (using the canonical translation of clauses into linear inequalities) in length $\mathrm{O}\big(L^2\big)$.

Hint: Show that cutting planes can efficiently simulate an application of the resolution rule on (the linear inequalities representing) a pair of clauses.

8b (10 p) Let G be an undirected connected graph with all vertices having even degree bounded by some constant. Associate a variable x_e with every edge $e \in E(G)$. For every vertex v, let $E(v)$ denote the set of edges incident to v. Let the *Even colouring formula* $EC(G)$ consist of the conjunction of the set of clauses

$$\bigwedge_{\substack{S \subseteq E(v) \\ |S| = |E(v)|/2+1}} \left(\left(\bigvee_{e \in S} x_e \right) \wedge \left(\bigvee_{e \in S} \overline{x}_e \right) \right)$$

for all vertices $v \in V(G)$ (encoding that the number of true and the number of false edges incident to vertex v are equal).

Prove that $EC(G)$ is unsatisfiable if and only if the number of edges $|E(G)|$ is odd.

8c (20 p) Prove that for any undirected connected graph G with all vertices having even degree bounded by some (universal) constant K and with an odd total number of edges it holds that cutting planes can refute $EC(G)$ efficiently. State explicitly what is the length of such an efficient cutting planes refutation that you can find. (You do not have to worry about superoptimizing it—any reasonable bound is fine—but it should match your refutation.)

Hint: Derive in cutting planes for each vertex v that the inequalities $\sum_{e \in E(v)} x_e \geq |E(v)|/2$ and $\sum_{e \in E(v)} -x_e \geq -|E(v)|/2$ must hold. What happens if you sum these inequalities over all $v \in V(G)$? If you wish, you may assume for simplicity (and without loss of any points) that all vertices v in G have degree at most 4.

8d (175 bonus points) To prove that cutting planes is exponentially stronger than resolution one would also show that resolution requires exponential length to refute $EC(G)$ if the graph G is well-connected enough. This is not really meant to be a problem on the problem set, but just for your information let us sketch what a formal claim could look like.

Say that G is a (d, s, e)-*edge expander graph* if all vertices have degree at most d and for all sets $S \subseteq V(G)$, $|S| \leq s$, it holds that $\left|E\left(S, \overline{S}\right)\right| = \left|\left\{(u, v) \in E(G) : u \in S, v \in \overline{S}\right\}\right| \geq e|S|$. Then for (d, s, e)-edge expanders G such that $s = \Omega\left(|V(G)|\right)$ and e can be chosen close enough to d—for instance, a randomly sampled 6-regular graph should probably do—it holds that $EC(G)$ requires exponential resolution length. To earn a lot of bonus points, you could work out the details to prove a formal statement along the lines above.

Remark: If you really want to attack this problem, talk to the main instructor first about what to read up on—there are better methods than prosecutor-defendant games.

9 (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.

10 (60 p) The goal of this exercise is to give a complete proof that $\mathsf{PSPACE} \subseteq \mathsf{IP}$, strengthening the result $\mathsf{coNP} \subseteq \mathsf{IP}$ 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 $\mathsf{coNP} \subseteq \mathsf{IP}$ 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 $\#\mathrm{SAT}_D$ problem for CNF formulas no longer works, since each multiplication corresponding to a $\forall$-quantifier can double the degree of the polynomial.

10a (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 $\mathrm{O}(n)$ since the degree blow-up is at most a constant factor 2.

10b (20 p) Assuming that any QBF formula ψ can be rewritten to satisfy the property in Problem 10a, use this to show that $\mathrm{TQBF} \in \mathsf{IP}$ and hence $\mathsf{PSPACE} \subseteq \mathsf{IP}$.

10c (20 p) Show that any QBF formula ψ of size m can be transformed into a logically equivalent formula ψ' of size $\mathrm{O}(m^2)$ that satisfies the property in Problem 10a.

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.