KTH Computer Science
and Communication

DD2446 Complexity Theory: Problem Set 4

Due: Friday November 8, 2013, at 23:59. Submit your solutions as a PDF file by e-mail to `jakobn at kth dot se` with the subject line `Problem set 4:` ⟨`your full name`⟩. Name the PDF file `PS4_`⟨`YourFullName`⟩`.pdf` (with your name coded in ASCII without national characters), and also 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. 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 solution individually and understand all aspects of it fully. You should also acknowledge any collaboration. State at the beginning of the problem set if you have been collaborating with someone and if so with whom.

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. It is hard to pin down 100% formal rules on what all 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 50 points should be enough for grade E, 80 points for grade D, 110 points for grade C, 140 points for grade B, and 170 points for grade A on this problem set. Any corrections or clarifications will be given at `piazza.com/kth.se/fall2013/dd2446/` and any revised versions will be posted on the course webpage `www.csc.kth.se/utbildning/kth/kurser/DD2446/kplx13/`.

1 (10 p) Suppose that $\mathcal{P}$ is a strongly testable property, where $\mathcal{P} = \bigcup_{n=1}^{\infty} \mathcal{P}_n$ for $\mathcal{P}_n \subseteq \{f : \mathsf{D}_n \to \mathsf{R}\}$ (i.e., some subset of all functions from some family of domains D_n to some fixed range R). Prove that there is a non-adaptive one-sided tester for $\mathcal{P}$ with constant query complexity.

2 (10 p) Prove that for any function $f : X \times Y \to Z$ it holds that $R_\epsilon^{priv}(f) \geq R_\epsilon^{pub}(f)$.

3 (20 p) Recall that an instance of the Max q-CSP$_W$ problem over n variables is a collection of m constraints $(C_1, I_1), (C_2, I_2), \ldots, (C_m, I_m)$, where each $C_j : [W]^q \to \{0,1\}$ is some q-ary predicate and each $I_j = \{i_{j,1}, i_{j,2}, \ldots, i_{j,q}\} \in [n]^q$ is a set of q variable indices. An assignment $\alpha \in [W]^n$ satisfies (C_j, I_j) if $C_j\bigl(\alpha_{i_{j,1}}, \alpha_{i_{j,2}}, \ldots, \alpha_{i_{j,q}}\bigr) = 1$, and the task is to compute the maximal number of constraints that can be satisfied by any assignment. Determine for which values of $q, W \in \mathbb{N}^+$ the Max q-CSP$_W$ problem is easy and for which values it is NP-hard.
Remark: Note that Max q-CSP$_W$ is not a decision problem, and so it does not quite make sense to ask whether it is NP-complete or not. However, we can still prove that it is NP-hard in the sense that there is a polynomial-time reduction from some NP-complete problem so that we could decide this problem efficiently if we had a polynomial-time algorithm for Max q-CSP$_W$.

4 (20 p) In the final lectures we proved a lower bound $R_\epsilon^{pub}(\textsc{ip}) = \Omega(n)$ for some fixed but small error probability ϵ on the randomized communication complexity of the inner product function $\textsc{ip}(x, y) = \sum_{i=1}^{n} x_i y_i \pmod 2$ for $x, y \in \{0, 1\}^n$. We also noted that if we allow error probability $\epsilon = 1/2$, then there is a trivial 1-bit protocol that just outputs a random guess (which has a 50% chance of being correct).

In this problem, we want to study what happens with $R_\epsilon^{pub}(\textsc{ip})$ when we relax the requirements on the protocol so that the error ϵ is not small but is allowed to approach $1/2$. Suppose that we let $\epsilon = 1/2 - \epsilon'$ for some small but constant ϵ', meaning that the protocol is only required to have a small ϵ'-advantage over random guessing. How does the communication complexity change? Are there still strong lower bounds? What happens for $\epsilon = 1/2 - \epsilon'(n)$ such that $\epsilon'(n) \to 0$ when $n \to \infty$, i.e., when the advantage over random guessing vanishes as n increases?

Determine for how small ϵ' (constant or subconstant) we can still obtain an $\Omega(n)$ lower bound for $R_{1/2-\epsilon'}^{pub}(\textsc{ip})$ using the techniques we covered in class. For full credit, you do not need to get additive or multiplicative constants exactly right, but the asymptotic bounds should be optimal.

5 (30 p) In Per's first guest lecture on the PCP theorem and hardness of approximation, we saw two versions of the PCP theorem. The first version was phrased as follows:

> *There is some universal constant $\delta > 0$ such that given a* Max 3-Sat *instance φ it is* NP*-hard to distinguish between the cases $Opt(\varphi) = 1$ and $Opt(\varphi) \leq 1 - \delta$.*

We then defined $\mathsf{PCP}_{c,s}[r, q, W]$ to be class of languages having probabilistically checkable proofs over an alphabet of size W with completeness c and soundness error s, where the verifier uses r random bits and makes q queries to the proof. This allowed us to give an equivalent formulation of the theorem as stated next:

$$\mathsf{NP} = \mathsf{PCP}_{1,1/2}[\mathrm{O}(\log n), \mathrm{O}(1), 2].$$

We proved in class that the first version of the PCP theorem implies the second, but only sketched the opposite direction.

Give a complete proof of the fact that the second version above of the PCP theorem implies the first version. (This will involve rephrasing some of what was said in class in your own words, but also filling in the details that were omitted during the lecture.)

6 (20 p) One of the two key technical lemmas in the Dinur's proof of the PCP theorem, which we outlined in class, was a *gap amplification lemma* which we stated (essentially) as follows:

> *For every $\ell \in \mathbb{N}^+$ there exist $W \in \mathbb{N}^+$, $\kappa > 0$ and $\epsilon_0 > 0$ such that there is a polynomial-time reduction R from* Max 2-CSP$_3$ *to* Max 2-CSP$_W$ *satisfying the following properties:*
>
> - *If $Val(\varphi) = 1$, then $Val(R(\varphi)) = 1$.*
> - *If $Val(\varphi) \leq 1 - \epsilon$ for $\epsilon < \epsilon_0$, then $Val(R(\varphi)) \leq 1 - \ell\epsilon$.*
> - $|R(\varphi)| \leq \kappa \cdot |\varphi|$.

In this lemma, the blow-up of the alphabet size is from 3 to some arbitrary W, but this W is independent of the size of the Max 2-CSP$_3$ instance φ.

Show that if we instead allow the alphabet blow-up to be a function of the instance size (and drop the condition that the reduction should be polynomial-size), then for every $\epsilon' > 0$ there exist $W = W(|\varphi|)$ such that there is a reduction R from Max 2-CSP$_3$ to Max 2-CSP$_W$ as above except that if $Val(\varphi) < 1$, then $Val(R(\varphi)) \leq \epsilon'$.

7 (40 p) For $x, y \in \{0,1\}^n$ interpreted as integers in $[0, 2^n - 1]$, let $\textsc{gt}(x,y) = [x > y]$ be the function that evaluates to 1 if $x > y$ and 0 otherwise.

7a (15 p) Determine exactly the deterministic communication complexity $D(\textsc{gt})$.

7b (25 p) What is the best upper bound you can give for $R^{pub}_{\epsilon}(\textsc{gt})$? (Say, for $\epsilon = 1/4$ if you want a concrete ϵ, but giving asymptotic bounds for any constant $\epsilon \in (0, 1/2)$ is also fine.)

8 (60 p) In Per's first lecture we also discussed how the class of languages captured by $\mathsf{PCP}_{c,s}[r, q, W]$ changes as we vary the parameters. For instance, we saw that $\mathsf{PCP}_{1,0}[0, \mathrm{poly}(n), 2] = \mathsf{NP}$ and that $\mathsf{PCP}_{1,0}[\mathrm{O}(\log n), 42, 2] = \mathsf{P}$. In this problem, we want to study this phenomenon further.

8a (30 p) Show that $\mathsf{PCP}_{1,2^{-10^{42}}}[\mathrm{poly}(n), 42, 2] = \mathsf{coRP}$.

8b (30 p) Can you give some argument why for larger values $s' > 2^{-10^{42}}$ of the soundness error it might hold that $\mathsf{PCP}_{1,s'}[\mathrm{poly}(n), 42, 2] \neq \mathsf{coRP}$ (under some more or less believable complexity-theoretic assumption, say)? How large do you need s' to be for this argument?

9 (90 p) The *falsified clause search problem* is the following communication problem. The starting point is some fixed unsatisfiable CNF formula F and some fixed partition $X \dot{\cup} Y = \mathit{Vars}(F)$ of the variables of F between Alice and Bob. Given as inputs truth value assignments $\alpha_X : X \to \{0,1\}$ and $\alpha_Y : Y \to \{0,1\}$, Alice and Bob should communicate to find a clause $C \in F$ that is falsified by the assignment $\alpha = \alpha_X \cup \alpha_Y$. (Such a clause always exists since F is unsatisfiable.)

The *pyramid graph* Π_h of height h is a DAG with $h+1$ layers, where there is one vertex in the highest layer (the sink z), two vertices in the next layer et cetera, down to $h+1$ vertices in the lowest layer 0. The ith vertex in layer L has incoming edges from the ith and $(i+1)$st vertices in layer $L-1$. See Figure 1(a) for an illustration of the pyramid graph of height 2.

The purpose of this problem is to investigate the hardness of the falsified clause search problems for certain CNF formulas defined in terms of pyramids.

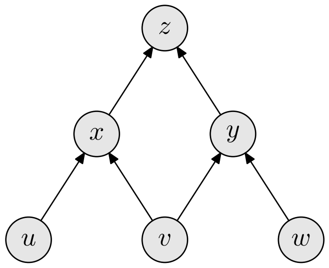
9a (30 p) The so-called *pebbling formula* over Π_h, denoted Peb_{Π_h}, is the conjunction of the following clauses:

- for all vertices s in the bottom layer, a unit clause s (i.e., a clause of size 1),
- For all vertices w in layers $L \geq 1$ with predecessors u, v, the clause $\overline{u} \lor \overline{v} \lor w$,
- for the sink z, the unit clause $\overline{z}$.

Figure 1(b) shows the formula corresponding to the pyramid in Figure 1(a).

Give the best upper and lower bounds you can for the deterministic communication complexity of the falsified clause search problem for Peb_{Π_h}. Your bounds should hold for any partition $X \dot{\cup} Y = \mathit{Vars}(\mathit{Peb}_{\Pi_h})$ of the variables. Express your bounds in terms of the number of vertices $n = (h+1)(h+2)/2$ in the graph Π_h. For full credit the bounds should be asymptotically tight.

Hint: Some kind of binary search might be a fruitful idea here.



(a) Pyramid graph Π_2 of height 2.

$$\begin{aligned} & u \\ \land\ & v \\ \land\ & w \\ \land\ & (\overline{u} \lor \overline{v} \lor x) \\ \land\ & (\overline{v} \lor \overline{w} \lor y) \\ \land\ & (\overline{x} \lor \overline{y} \lor z) \\ \land\ & \overline{z} \end{aligned}$$

(b) Pebbling formula Peb_{Π_2}.

Figure 1. Example pebbling formula.

9b (20 p) In the *XORified pebbling formula* $Peb_{\Pi_h}[\oplus]$ over Π_h we instead think of each vertex v as the exclusive or of two variables $v_1 \oplus v_2$ and have the following clauses:

- for all vertices s in the bottom layer, the CNF encoding of $s_1 \oplus s_2$,
- For all w in layers $L \geq 1$ with predecessors u, v, the CNF encoding of $\neg(u_1 \oplus u_2) \lor \neg(v_1 \oplus v_2) \lor (w_1 \oplus w_2)$,
- for the sink z, the CNF encoding of $\neg(z_1 \oplus z_2)$.

Figure 2 shows the XORified pebbling formula for the pyramid in Figure 1(a).

Give the best upper bound you can for the deterministic communication complexity of the falsified clause search problem for $Peb_{\Pi_h}[\oplus]$, where Alice gets all variables $X = \{u_1, v_1, w_1, \ldots\}$ and Bob gets all variables $Y = \{u_2, v_2, w_2, \ldots\}$ (again expressed in terms of the number of vertices $n = (h+1)(h+2)/2$ in Π_h).

9c (40 p) We say that a randomized protocol $\mathcal{P}$ solves the falsified clause search problem for F *consistently* if $\mathcal{P}$ computes some function $f : \mathit{Vars}(F) \to \{C \in F\}$ except with error probability ϵ, where f is such that for any input $\alpha = \alpha_X \cup \alpha_Y$ it holds that $f(\alpha) = C_\alpha$ is a clause in F falsified by α.

Observe that deterministic protocols are consistent by definition, but for assignments α falsifying several clauses in F a randomized protocol could potentially give different answers depending on the random coin flips. For a consistent protocol, we require that it always outputs some specific falsified clause $f(\alpha) = C_\alpha$ except with probability ϵ, and any other clause is considered to be an error even if it is falsified by α. Note, however, that the protocol can choose any function f it wants, and in particular is free to pick any falsified clause C_α for each α independent of all other choices.

What is the cost of the best consistent randomized protocol you can give for $Peb_{\Pi_h}[\oplus]$ with variables partitioned as in Problem 9b, expressed in terms of $n = \Theta(h^2)$?

9d (∞ p) Prove a $\omega(\log n)$ deterministic communication complexity lower bound for the falsified clause search problem for $Peb_{\Pi_h}[\oplus]$ with variables partitioned as in Problem 9b (ideally a bound on the form $\Omega(n^\delta)$ for some $\delta > 0$), or establish that no such lower bound exist.

Remark: This is an open research problem, and so you are not necessarily expected to solve it...

$$
\begin{aligned}
& (u_1 \lor u_2) \\
\land\ & (\overline{u}_1 \lor \overline{u}_2) \\
\land\ & (v_1 \lor v_2) \\
\land\ & (\overline{v}_1 \lor \overline{v}_2) \\
\land\ & (w_1 \lor w_2) \\
\land\ & (\overline{w}_1 \lor \overline{w}_2) \\
\land\ & (u_1 \lor \overline{u}_2 \lor v_1 \lor \overline{v}_2 \lor x_1 \lor x_2) \\
\land\ & (u_1 \lor \overline{u}_2 \lor v_1 \lor \overline{v}_2 \lor \overline{x}_1 \lor \overline{x}_2) \\
\land\ & (u_1 \lor \overline{u}_2 \lor \overline{v}_1 \lor v_2 \lor x_1 \lor x_2) \\
\land\ & (u_1 \lor \overline{u}_2 \lor \overline{v}_1 \lor v_2 \lor \overline{x}_1 \lor \overline{x}_2) \\
\land\ & (\overline{u}_1 \lor u_2 \lor v_1 \lor \overline{v}_2 \lor x_1 \lor x_2) \\
\land\ & (\overline{u}_1 \lor u_2 \lor v_1 \lor \overline{v}_2 \lor \overline{x}_1 \lor \overline{x}_2) \\
\land\ & (\overline{u}_1 \lor u_2 \lor \overline{v}_1 \lor v_2 \lor x_1 \lor x_2) \\
\land\ & (\overline{u}_1 \lor u_2 \lor \overline{v}_1 \lor v_2 \lor \overline{x}_1 \lor \overline{x}_2) \\
\land\ & (v_1 \lor \overline{v}_2 \lor w_1 \lor \overline{w}_2 \lor y_1 \lor y_2) \\
\land\ & (v_1 \lor \overline{v}_2 \lor w_1 \lor \overline{w}_2 \lor \overline{y}_1 \lor \overline{y}_2)
\end{aligned}
$$

$$
\begin{aligned}
\land\ & (v_1 \lor \overline{v}_2 \lor \overline{w}_1 \lor w_2 \lor y_1 \lor y_2) \\
\land\ & (v_1 \lor \overline{v}_2 \lor \overline{w}_1 \lor w_2 \lor \overline{y}_1 \lor \overline{y}_2) \\
\land\ & (\overline{v}_1 \lor v_2 \lor w_1 \lor \overline{w}_2 \lor y_1 \lor y_2) \\
\land\ & (\overline{v}_1 \lor v_2 \lor w_1 \lor \overline{w}_2 \lor \overline{y}_1 \lor \overline{y}_2) \\
\land\ & (\overline{v}_1 \lor v_2 \lor \overline{w}_1 \lor w_2 \lor y_1 \lor y_2) \\
\land\ & (\overline{v}_1 \lor v_2 \lor \overline{w}_1 \lor w_2 \lor \overline{y}_1 \lor \overline{y}_2) \\
\land\ & (x_1 \lor \overline{x}_2 \lor y_1 \lor \overline{y}_2 \lor z_1 \lor z_2) \\
\land\ & (x_1 \lor \overline{x}_2 \lor y_1 \lor \overline{y}_2 \lor \overline{z}_1 \lor \overline{z}_2) \\
\land\ & (x_1 \lor \overline{x}_2 \lor \overline{y}_1 \lor y_2 \lor z_1 \lor z_2) \\
\land\ & (x_1 \lor \overline{x}_2 \lor \overline{y}_1 \lor y_2 \lor \overline{z}_1 \lor \overline{z}_2) \\
\land\ & (\overline{x}_1 \lor x_2 \lor y_1 \lor \overline{y}_2 \lor z_1 \lor z_2) \\
\land\ & (\overline{x}_1 \lor x_2 \lor y_1 \lor \overline{y}_2 \lor \overline{z}_1 \lor \overline{z}_2) \\
\land\ & (\overline{x}_1 \lor x_2 \lor \overline{y}_1 \lor y_2 \lor z_1 \lor z_2) \\
\land\ & (\overline{x}_1 \lor x_2 \lor \overline{y}_1 \lor y_2 \lor \overline{z}_1 \lor \overline{z}_2) \\
\land\ & (z_1 \lor \overline{z}_2) \\
\land\ & (\overline{z}_1 \lor z_2)
\end{aligned}
$$

Figure 2. XORified pebbling contradiction $Peb_{\Pi_2}[\oplus]$.