KTH Computer Science and Communication

DD2445 Complexity Theory: Problem Set 4

Submission: Due *Monday January 22, 2018, 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 4: ⟨your full name⟩`. Name the PDF file `PS4_⟨YourFullName⟩.pdf` with your name written in CamelCase without blanks and in ASCII without national characters. 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 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 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. 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 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 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.

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 100 points will be enough for grade E, 140 points for grade D, 180 points for grade C, 220 points for grade B, and 260 points for grade A on this problem set. Any corrections or clarifications will be given at `piazza.com/kth.se/fall2017/dd2445/` and any revised versions will be posted on the course webpage `www.csc.kth.se/DD2445/kplx17/`.

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 in Lectures 21 and 22, 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 notes for Lecture 21).

Suppose that we would instead have been given the circuit lower bound stated in Theorem 4 in the notes for Lecture 20 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 (40 p) Let F be an unsatisfiable CNF formula and let α denote any truth value assignment to the variables in F. The *search problem* for F given α is to find a clause $C \in F$ falsified by α.

A *decision tree* T_F for F is a binary tree with leaves labelled by clauses in F, internal vertices labelled by variables x, and two edges from each internal vertex labelled 0 and 1. Any α defines a path through T_F starting from the root, following from each internal vertex x the edge labelled by the value $\alpha(x)$, and ending in some leaf C that is the *answer of* T_F *on* α. The tree T_F *solves the search problem for* F if on any α the answer C is a clause falsified by α.

Let us write $S_D(F)$ to denote the minimal size (i.e., number of vertices) of any decision tree solving the search problem for F, and write $L_\tau(F \vdash \bot)$ to denote the minimal length of any tree-like resolution refutation of F. Your task is to prove that decision trees and tree-like resolution refutations are essentially just two different ways of describing the same objects.

Please note that Problems 4a, 4b, and 4c below can be solved independently of one another and that results from preceding subproblems can be used in succeeding subproblems regardless of which problems were actually solved.

4a (10 p) Prove that $S_D(F) \leq L_\tau(F \vdash \bot)$ by showing that any tree-like resolution refutation of F can be made into a decision tree solving the search problem for F.

4b (20 p) Prove that $L_\tau(F \vdash \bot) \leq S_D(F)$ by showing that any decision tree solving the search problem for F can be made into a tree-like resolution refutation of F.

4c (10 p) Argue that this proves the implicational completeness of resolution, and, in particular, shows that any unsatisfiable CNF formula over n variables has a resolution refutation π in length $L(\pi) = \exp(\mathrm{O}(n))$. What is the best concrete bounds you can get, not using big-oh notation but providing explicit constants instead?

5 (40 p) In the monotone circuit lower bound for clique presented in Lectures 19 and 20 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 notes for Lecture 21 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 $\mathrm{O}(L)$.

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.

7 (90+ p) Given an undirected graph $G = (V,E)$ and a parameter $k \in \mathbb{N}^+$, for the purposes of this and the following problems let us define the formula $\mathit{Clique}(G,k)$ encoding the claim that G has a k-clique as consisting of the following clauses:

$$\bigvee_{v \in V} x_{v,i} \qquad i \in [k]; \tag{1}$$

$$\overline{x}_{u,i} \vee \overline{x}_{v,i} \qquad i \in [k] \text{ and } u,v \in V,\ u \neq v; \tag{2}$$

$$\overline{x}_{u,i} \vee \overline{x}_{v,j} \qquad i,j \in [k],\ i \neq j, \text{ and } u,v \in V,\ (u,v) \notin E. \tag{3}$$

Hint: In what follows, you are encouraged to use the fact that tree-like resolution refutations of a formula F are equivalent to decision trees for the falsified clause search problem for F, and that regular resolution refutations are equivalent to read-once branching programs.

7a (40 p) Consider formulas $\mathit{Clique}(G_n,k)$ defined over complete $(k-1)$-partite graphs, i.e., graphs over $n = (k-1)n'$ vertices with $V = V_1 \dot\cup V_2 \dot\cup \cdots \dot\cup V_{k-1}$ for $|V_i| = n'$ and with edge set $E = \bigcup_{1 \leq i < j \leq k-1} \{(v_i,v_j) \mid v_i \in V_i, v_j \in V_j\}$. A moment of thought reveals that such graphs do not contain k-cliques. Prove that tree-like resolution proofs require length $n^{\Omega(k)}$ to establish this fact.

7b (50 p) Prove that formulas $\mathit{Clique}(G_n,k)$ defined over complete $(k-1)$-partite graphs are easy to refute for regular resolution in that they only require refutations of length $n^{\mathrm{O}(1)}$.

7c (300+ p) **Open problem:** Find some family of graphs $\{G_n\}_{n=1}^{\infty}$ on n vertices that do not contain k-cliques but are such that the refutation length of formulas $\mathit{Clique}(G_n, k)$ in general resolution grows like $n^{\Omega(k)}$, or indeed like $n^{\omega_k(1)}$ for any arbitrarily slowly growing but unbounded function $\omega_k(1)$ of k. If it helps, you can also omit the clauses (2) from the formulas (removing these clauses can only make the formulas harder to refute, although it is clear that they are still unsatisfiable).

8 (40 p) With notation and terminology as in the notes for Lecture 23, let $\mathcal{D}$ denote the random distribution over paths that we defined for any arbitrary but fixed read-once branching program solving the falsified clause search problem for a given clique formula. Prove Observation 3 by establishing the claims in the following two subproblems.

8a Any path $\alpha \sim \mathcal{D}$ ends by ruling out some index i (i.e., it never falsifies a functionality or edge axiom).

8b Any path $\alpha \sim \mathcal{D}$ sets at most k variables to 1.

9 (50 p) Your task in this problem is to prove Observation 5 in the notes for Lecture 23. With notation and terminology as in the lecture notes, suppose that a and b are nodes in a read-once branching program and that α is a path that passes through a and b (in that order) and ends by ruling out clique membership index i. Define

$$V^i_{a,b}(\alpha) = \left\{ v \in V \,\middle|\, x_{v,i} \text{ is set to 0 in } \alpha \text{ between } a \text{ and } b \right\} .$$

Show that any path α' passing through a and b sets $x_{v,i}$ to 0 precisely for all vertices $v \in V^i_{a,b}(\alpha)$ between a and b, i.e., that the vertex set $V^i_{a,b}(\alpha') = V^i_{a,b}(\alpha)$ is independent of the path and only depends on the branching program nodes a and b.

10 (60 p) Your task in this problem is to prove Lemma 9 in Lecture 23. With notation and terminology as in the notes for this lecture, suppose that $G = (V, E)$ is a (k, r, s, ϵ)-clique-dense graph. Show that for any $\alpha \sim \mathcal{D}$ it holds with probability 1 that there is a pair of nodes (a, b) in the read-once branching program such that α usefully traverses (a, b).

Hint: Use the neighbour-denseness of V and Observation 5 above.