KTH Computer Science and Communication

DD2442 Proof Complexity: Problem Set 1

Due: Wednesday September 28, 2016, 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 1:` ⟨`your full name`⟩. Name the PDF file `PS1_`⟨`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 (at least 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 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 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. On the contrary, you can chooose to solve just a subset of the problems and still get a top grade. A total score of around 70 points should be enough for grade E, 105 points for grade D, 140 points for grade C, 175 points for grade B, and 210 points for grade A on this problem set. Any corrections or clarifications will be given at `piazza.com/kth.se/fall2016/dd2442/` and any revised versions will be posted on the course webpage `www.csc.kth.se/DD2442/semteo16/`.

1 (10 p) In the very first lecture, we decided to focus only on CNF formulas and said that this is essentially without loss of generality since any propositional logic formula F can be transformed to CNF formula F' such that F' only linearly larger than F and is unsatisfiable iff F is a tautology. We did do the full transformation, however, and your task now is to fill in the missing details regarding the connectives $\wedge$ and $\leftrightarrow$.

Given a formula $F \doteq G \wedge H$, show how to write CNF clauses that force x_F to take the value of $G \wedge H$ assuming that x_G and x_H are correctly representing the truth values of G and H, respectively. Then solve the same problem for the formula $F \doteq G \leftrightarrow H$. Please do not forget to argue why your encodings into CNF are correct.

2 (20 p) The purpose of this problem is to clarify the relation between edge expanders and connectivity expanders as defined in the third lecture. In what follows below, all graphs $G = (V, E)$ are assumed to be connected, and δ and c are constants independent of the size of the graph.

2a (10 p) Prove that if $G = (V, E)$ is a (d, δ)-edge expander, then G is also a (d, c)-connectivity expander for $c \geq \delta/4$.

Hint: Consider a minimal edge set E' that disconnects G into components of size at most $|V|/2$ and reason about the edge expansion of these components.

2b (10 p) Prove that there is some constant $c > 0$ such that for any $\delta > 0$ it holds that for all large enough $n \in \mathbb{N}^+$ there is an n-vertex (d, c)-connectivity expander that is not a (d, δ)-edge expander.

3 (20 p) When proving theorems about the resolution proof system it is sometimes technically convenient to also have a second inference rule, namely the *weakening rule* that allows to derive $C \vee D$ from the clause C (where D can be any arbitrarily chosen clause). Since $C \vee D$ is a weaker clause than C it seems intuitively clear that this should not be a very useful rule, and indeed it is the case that any use of weakening in a resolution refutation can be eliminated without loss of generality. Your task is to prove this formally.

That is, prove that if $\pi : F \vdash \bot$ is a resolution refutation with weakening, then there is another resolution refutation $\pi' : F \vdash \bot$ in at most the same length that does not use the weakening rule.

Hint: Do a proof by forward induction over the resolution refutation $\pi = (C_1, C_2, \ldots, C_L)$ containing weakening steps.

4 (30 p) In the Prosecutor-Defendant game for PHP formulas, say that Prosecutor has a *size-L strategy for winning against a class of Defendant strategies* $\mathcal{D}$ if there is an instruction book with at most L records such that Prosecutor can always win by using this instruction book when Defendant uses any strategy $D \in \mathcal{D}$. (Thus, a *complete Prosecutor strategy* is a strategy for winning against the class $\mathcal{D}_{\text{all}}$ of all possible Defendant strategies, but here we are interested also in incomplete Prosecutor strategies.)

4a (15 p) Let $\mathcal{D}$ be the set consisting of the single defendant strategy which always gives the answer "no" to the question "Does pigeon i fly to hole j?" as long as there is some other hole permitted for pigeon i according to Prosecutor's current record (and otherwise answers "yes" if the forced choice for pigeon i is pigeonhole j). Show that Prosecutor has a strategy in size $\mathrm{O}(n)$ for winning when Defendant plays according to this strategy.

4b (15 p) Let $\mathcal{D}$ be the set consisting of all the defendant strategies of picking uniformly at random n out of the $n+1$ pigeons and matching them randomly to pigeonholes, and then always answer consistently with this randomly chosen matching. Show that Prosecutor has a strategy in size $\mathrm{O}(n^2)$ for winning when Defendant plays according to this strategy.

5 (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 $L_{\mathcal{T}}(F \vdash \bot)$ to denote the minimal length of any tree-like resolution refutation of F. A very convenient fact is that decision trees and tree-like resolution refutations are essentially just two different ways of looking at the same object. Your task is to formalize this claim as described below.

Please note that Problems 5a, 5b, and 5c 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.

5a (10 p) Prove that $S_D(F) \leq L_{\mathcal{T}}(F \vdash \bot)$ by showing that any tree-like resolution refutation of F can be made into a decision tree solving the search problem for F.

5b (20 p) Prove that $L_{\mathcal{T}}(F \vdash \bot) \leq S_D(F)$ by showing that any decision tree solving the search problem for F can be made into a tree-like resolution refutation of F. (For partial credit, just prove $L_{\mathcal{T}}(F \vdash \bot) = \mathrm{O}(S_D(F))$ using the weakening rule and the fact claimed in Problem 3, which you can use regardless of whether you solved that problem or not.)

5c (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?

6 (60 p) Using notation and terminology from the second lecture, let I_R denote the set of thoroughly investigated pigeons in a Prosecutor record and assume that R is *informative*, i.e., that $|I_R| \geq n/4$. We claimed that when Defendant randomly chooses a partial matching of $n/4$ pigeons, then the probability that the size of the intersection is less than $n/32$ is at most exponentially small, i.e., at most $2^{-\epsilon n}$ for some $\epsilon > 0$ for n large enough. Your task is to perform the calculations to prove this. In what follows we assume tacitly, but without loss of generality, that n is large enough for our claims to hold and also for simplicity that 32 evenly divides n.

Please note that Problems 6a, 6b, and 6c 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.

6a (10 p) Explain, briefly but clearly and convincingly, why the probability that the size of the intersection is less than $n/32$ is at most

$$\frac{\sum_{i=0}^{n/32-1} \binom{n/4}{i}\binom{n+1-n/4}{n/4-i}}{\binom{n+1}{n/4}} \,.$$

6b (20 p) Show that

$$\sum_{i=0}^{n/32-1} \binom{n/4}{i}\binom{n+1-n/4}{n/4-i} \leq \frac{n}{32}\binom{n/4}{n/32}\binom{3n/4+1}{7n/32} \ .$$

6c (30 p) Using the inequalities in Problems 6a and 6b as well as *Stirling's formula*

$$\sqrt{2\pi m}\left(\frac{m}{e}\right)^m e^{\frac{1}{12m+1}} < m! < \sqrt{2\pi m}\left(\frac{m}{e}\right)^m e^{\frac{1}{12m}} \ ,$$

prove that the probability of a small intersection is at most $2^{-\delta n}$ for some $\delta > 0$. (You should not have to be too careful with the calculations here—using Stirling to get the size of the main protagonists mostly right should be enough.)

7 (70 p) The purpose of this problem is to establish Lemma 9 in the notes from Lecture 3, i.e., that if $G = (V, E)$ is a connected graph with odd-charge function $\chi : V \to \{0, 1\}$ for which we randomly sample a charge-preserving assignment to a medium-large edge set $E_1 \subseteq E$, then for any subset $E_2 \subseteq E_1$ such that $G_2 = (V, E \setminus E_2)$ is connected this random sampling yields independent and uniformly random bits.

Please note that Problems 7a, 7b, and 7c 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.

7a (20 p) Let $A \subseteq \{0, 1\}^m$ be an affine subspace and suppose for a subset of coordinates $S \subseteq [m]$ that all bit strings in $\{0, 1\}^S$ are supported by A (i.e., for any $\beta \in \{0, 1\}^S$ there is a vector $u_\beta \in A$ that agrees with β on the coordinates in S). Prove that a uniformly random sample from A restricted to S yields independent and uniformly random bits.

Hint: There are a couple of suggested approaches for this problem in the lecture notes.

7b (20 p) Fix a connected graph $G = (V, E)$ with an arbitrary charge function $\chi : V \to \{0, 1\}$. Let $E' \subseteq E$ be any minimal set disconnecting G into exactly two connected graphs G_1 and G_2. Prove that for G and χ fixed as above, it holds for any assignment ρ of values in $\{0, 1\}$ to the edge variables in $\{x_e \mid e \in E'\}$ that whether G_1 and G_2 gets odd or even total charges only depends on the parity of the sum $\sum_{e \in E'} \rho(x_e)$.

Note that when we apply ρ to the edges in E', then χ is updated to χ' according to edge values in ρ so that $\chi'(v) = \chi(v) + \sum_{e \ni v, e \in E'} \rho(x_e) \pmod 2$. The total charge of G_i for $i \in \{1, 2\}$ is then defined as $\sum_{v \in V(G_i)} \chi'(v)$.

7c (30 p) Suppose that $G = (V, E)$ is a (d, δ)-edge expander for $\delta > 0$, that $E_1 \subseteq E$ is a moderate-size set such that $G_1 = (V, E \setminus E_1)$ has a unique connected component of size larger than $|V|/2$, and that $E_2 \subseteq E_1$ is such that $G_2 = (V, E \setminus E_2)$ is connected. Use the claims in Problems 7a and 7b together with the material in the "appendix notes" for Lecture 3 (which does not have to be reproven, but state clearly what you use and how) to prove that if we sample uniformly at random a charge-preserving assignment ρ to E_1, then the values assigned to the edges in E_2 are independent and uniformly random bits.

$$\begin{pmatrix}
1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & \mathbf{1} & 0 & 0 & 1
\end{pmatrix}$$

(a) Matrix encoding row and column constraints.

$$\begin{aligned}
&(x_{1,1} \vee x_{1,2} \vee x_{1,4}) \\
\wedge\ &(x_{1,1} \vee x_{1,2} \vee x_{1,8}) \\
\wedge\ &(x_{1,1} \vee x_{1,4} \vee x_{1,8}) \\
\wedge\ &(x_{1,2} \vee x_{1,4} \vee x_{1,8}) \\
&\ \ \vdots \\
\wedge\ &(\overline{x}_{4,11} \vee \overline{x}_{8,11} \vee \overline{x}_{10,11}) \\
\wedge\ &(\overline{x}_{4,11} \vee \overline{x}_{8,11} \vee \overline{x}_{11,11}) \\
\wedge\ &(\overline{x}_{4,11} \vee \overline{x}_{10,11} \vee \overline{x}_{11,11}) \\
\wedge\ &(\overline{x}_{8,11} \vee \overline{x}_{10,11} \vee \overline{x}_{11,11})
\end{aligned}$$

(b) Example cardinality constraints in CNF.

Figure 1: Matrix and (fragment of) corresponding subset cardinality formula in Problem 9.

8 (90+ p) A slightly annoying aspect of the Tseitin formula lower bound we did in Lecture 3 was that we only obtained lower bounds for (d, δ)-edge expanders with $\delta > 1$. It is known that the theorem is true for any $\delta > 0$. In this problem, we want to take a closer look at how the result proven in class can (or cannot) be improved if we assume that the graph is d-regular (i.e., that every vertex has exactly d incident edges).

8a (40 p) For partial credit on this subproblem, prove that refuting Tseitin formulas over random 4-regular graphs requires exponential length in resolution asymptotically almost surely (a.a.s). You might want to use that a.a.s. such graphs have edge expansion at least 0.4.

For full credit, prove an exponential lower bound (not almost surely, but unconditionally) for any 4-regular graph that has arbitrarily small but positive (and fixed) edge expansion $\delta > 0$.

8b (50 p) Can the approach in Lecture 3 be implemented to yield strong lower bounds for edge expanders with arbitrary expansion $\delta > 0$ if we just work a bit harder on the analysis? Or can you show that some fact/claim/proposition/lemma in the notes is provably false for positive but sufficiently small δ, $0 < \delta \leq 1$?

8c (100+ p) **Open problem:** Can you improve the Defendant lower bound strategy in Lecture 3 to prove for 3-regular graphs with arbitrarily small but positive (and fixed) edge expansion $\delta > 0$ that Tseitin formulas over such graphs require exponential resolution refutation length? This is known to be true, but the main instructor does not know of any way of using Prosecutor-Defendant games to prove such a lower bound.

9 This problem is about the subset cardinality formulas discussed in the first lecture. Please study the example in Figure 1 to understand what these formulas are encoding—this might be helpful when dealing with the more abstract, and perhaps harder to parse, description below.

Recall that we have an $n \times n$ 0/1-matrix M with exactly 4 ones per row and column, except that we add an extra one somewhere so that one row and one column has 5 ones. The variables of the formula are $x_{i,j}$ corresponding to positions in the matrix $M = (m_{i,j})$ such that $m_{i,j} = 1$. Every row provides a constraint that a majority of the variables in that row should be true. Every column requires that a majority of the variables in that column should be false.

In more formal notation, let $R_i = \{j \mid m_{i,j} = 1\}$ be the column indices for 1-entries in row i and let $C_j = \{i \mid m_{i,j} = 1\}$ be the row indices for 1-entries in column j. (We note in passing that specifying only the row sets R_i, $i \in [n]$, or only the column sets C_j, $j \in [n]$, is sufficient to describe M completely.) Then the subset cardinality formula $SC(M)$ corresponding to a matrix M contains the following clauses:

- For every row i the set of clauses $\left\{\bigvee_{j \in R^*} x_{i,j} \,\middle|\, R^* \subseteq R_i, |R^*| = 3\right\}$.
- For every column j the set of clauses $\left\{\bigvee_{i \in C^*} \overline{x}_{i,j} \,\middle|\, C^* \subseteq C_j, |C^*| = 3\right\}$.

It has been shown that the formula $SC(M)$ is exponentially hard for resolution if M is an expanding matrix (meaning, roughly, that every small-to-medium-large set of rows contain 1-entries in many different columns), and, in particular, the exponential lower bound holds asymptotically almost surely if M is a randomly sampled matrix subject to the constraints described above.

Please note that Problems 9a, 9b, and 9c below can be solved independently of one another.

9a (20 p) Prove that any subset cardinality formula $SC(M)$, regardless of any expansion properties of the matrix M, is always easy for cutting planes. Describe the structure of a short CP refutation and analyze how short you can get it to be. You do not actually have to write down every single low-level syntactic detail, but the description should be detailed enough so that a fellow student could purely mechanically reproduce the cutting planes refutation from your descripion without having to do any creative thinking.

9b (40 p) The example subset cardinality formula in Figure 1 has a very particular structure in that the first row contains 1s in positions 2^t for $t = 0, 1, 2, 3$ and that subsequent rows just has this pattern shifted down the diagonal and wrapping around when reaching the final column. In formal notation, we have $R_i = \left\{1 + (i + 2^t - 2 \bmod n) \,\middle|\, t = 0, 1, 2, 3\right\}$ (except that this ignores where the extra 1-entry is added, but exactly where this is done is not too important).

Prove that for such a regular matrix M which can be described by a pattern shifted down the diagonal (except for the extra 1 that appears somewhere), i.e., where there are a_i, $i \in [4]$, with $1 \leq a_1 < a_2 < a_3 < a_4 \leq K$ for some constant K independent of n, such that $R_i = \left\{1 + (i + a_t - 2 \bmod n) \,\middle|\, t \in [4]\right\}$, the subset cardinality formulas $SC(M)$ are in fact easy also for resolution.

Hint: Notice that, as usual, the above is an asymptotic claim, so you might need to pick n large enough for the upper bound to kick in (but not ridiculously large). There are actually resolution refutations in length only linear in the size of the formula, but polynomial length is sufficient to get a full score on this subproblem.

9c (60 p) An intriguing fact is that even for regular matrices M as in Problem 9b, for which subset cardinality formulas $SC(M)$ are theoretically very easy for resolution, the formulas can be really, really hard in practice for state-of-the-art so-called *conflict-driven clause learning (CDCL)* SAT solvers based on resolution. Moreover, the hardness seems to depend in subtle ways on the concrete patterns used, and we do not really understand this dependence. (This is a polite way of saying that there are some theoretical predictions regarding which patterns should be easy or hard, but somewhat interestingly these predictions sometimes seem to flatly contradict what can actually be observed in practice...)

The purpose of this final subproblem is to do some empirical research to come up with the hardest pattern you can find for very regular subset cardinality formulas $SC(M)$ as described above. This involves picking four numbers a_1, a_2, a_3, a_4, where we require $1 \leq a_1 < a_2 < a_3 < a_4 \leq 9$ to capture the hardness of small, compact patterns, adding an extra 1 somewhere, generating the corresponding subset cardinality formulas $SC(M)$, and running the SAT solver *MiniSat* on them (available in the Ubuntu environment at KTH CSC; see also the webpage `minisat.se`).

To solve this subproblem you will need to do the following:

1. Write code that given n and $1 \leq a_1 < a_2 < a_3 < a_4 \leq 9$ generates a file where the first line contains the number n repeated twice, to give the dimensions of the matrix, and all following lines are rows in the matrix, one row per line, where the specified pattern appears shifted along the diagonal (and with 0s and 1s separated by blanks). Note that you also need to flip a 0 to an extra 1 somewhere.
2. Write code for genering the subset cardinality formula $SC(M)$ from any matrix M given in the format described above. The formula should be in the standard *DIMACS* format described, e.g., at `www.csc.kth.se/DD2442/semteo16/useful-info`, where you can also find some information about *MiniSat*.

   Alternatively, you can install and use the tool *CNFgen* (available at `github.com/MassimoLauria/cnfgen`) to generate the formulas from matrices for which you coded up in item 1. Assuming that you have stored your matrix in the file `matrix.txt`, you can call *CNFgen* to write the corresponding formula to the file `formula.cnf` by doing `cnfgen subsetcard -gf matrix -i matrix.txt > formula.cnf` (assuming Unix-style file redirections).
3. Report which is the hardest pattern $1 \leq a_1 < a_2 < a_3 < a_4 \leq 9$ that you can find. Evaluate the hardness of a pattern by finding (and reporting) the smallest n such that *MiniSat* does not solve the formula $SC(M)$ for a matrix M of dimensions $n \times n$ generated from this pattern within 10 minutes of CPU time when run on the computer `u-shell.csc.kth.se`. Measure time as reported by the `time` command. There will be an extra bonus to the student who finds the very hardest pattern (with reproducible results when we run the same experiments on `u-shell`).

   Just to give a sense of the scale, you should expect these formulas to start getting seriously difficult for 25×25 matrices or so, i.e., for around 100 variables, or even earlier depending on how devious patterns you can design.