Satisfiability of Boolean Formulas - Combinatorics and Algorithms
Autumn 2008

Special Exercises

Regulations:

- This is the fourth and last exercise sheet.
- You are expected to solve it carefully and then write a nice and complete exposition of your solutions using LaTeX.
- You are welcome to discuss the tasks with your colleagues, but we expect each of you to hand in your own, individual writeup.
- Your solutions will be graded. The three highest out of your four achieved grades will account for 10% of your final grade for the course each (so 30% of the grade in total).

Special Exercise Set 4

Due date: Friday, December 5, 2008 (at the beginning of the 10:15 lecture)

Problem 1 The ppz k-SAT algorithm

Let F_k be the k-SAT formula that describes the parity function on variable set $\{x_1, x_2, \ldots, x_k\}$, i.e. F_k is satisfied by an assignment iff the number of variables set to true is odd.

a. Prove: ppz(F_k) outputs a satisfying assignment with probability 1.

Let now F be any satisfiable CNF formula which is closed under resolution, that is the resolvent of any two clauses in F is contained in F.

b. Prove: $\text{ppz}(F)$ outputs a satisfying assignment with probability 1.

Problem 2 Killing all clauses

Let F be any CNF formula over a variable set V.

Let $S \subseteq \{0,1\}^V$ be some set of assignments. We say that S is a *killing set* for F if for every clause $C \in F$, there exists an assignment $\alpha \in S$ such that α violates C.

Let $G_F = (F, E)$ be the graph in which the clauses of F constitute the vertices and an edge $\{C_i, C_j\} \in E$ exists between two clauses if they have at least one pair of complementary literals, i.e. there exists some literal $u \in C_i$ with $\bar{u} \in C_j$. Then G_F is said to be the *conflict graph* of F.

For any graph G, the *chromatic number* of G, usually denoted by $\chi(G)$, is the minimum number of colours necessary to properly colour the vertices of G.

Prove: the smallest killing set for F has size exactly $\chi(G_F)$.

HINT: You might want to consider the whole setting in the V-cube. Moreover, a result from the last special assignment set might come handy.

Exercises 3 and 4 are about the reading assignment
"A constructive proof of the Lovász Local Lemma".

Problem 3 Too simple a proof

Following is a (wrong!) 'proof' for the efficiency of the algorithm presented in the paper and it is your task to discover why it is wrong. The 'proof' circumvents the introduction of assignment tables and composite witnesses.

Proof sketch. Assume we run the randomized algorithm and at some point it has to abort because more than $\log(m) + 2$ recursive invocations have been made. We want to prove that this happens with probability less than $1/2$. Consider any fixed recursion tree τ of size exactly $u \in N$ and let $C_1, C_2, \ldots, C_u$ be its clauses in the natural ordering. The probability that C_1 was violated when we started is 2^{-k} because all assignments to its variables have been selected u.a.r. The probability that C_2 was violated in the next step is again 2^{-k} for the same reason. In total, the probability that the succession $C_1, C_2, \ldots, C_u$ of invocations occurs is 2^{-ku}. Since there are in total no more than $m(ed)^u < m2^{(k-1)u}$ many recursion trees of size u (by the usual argument), the probability that one of them having size at least $\log(m) + 2$ occurs is smaller than $1/2$.

- **a.** Argue in as much detail as possible why a proof along these lines is not correct.
- **b.** Suppose we have a formula F such that for all $C \in F$, the weaker bound $|\Gamma_F^+(C)| \leq 2^{k-3}$ holds. Prove that with probability at least $1/2$, the *first* invocation of locally_correct(F,...) will return timely, without being interrupted.

Problem 4 Fewer trees

Let F be a formula as required in Theorem 1.2. Let us say that a composite witness W is *valid* if it can occur during execution of the algorithm, that is if there exists a table of assignments such that running the algorithm on that table will produce a collection of recursion trees of which W is a maximal connected component (connected in the dependency graph, as usual).

a. Explain why not all composite witnesses are valid.

b. Prove that in each recursion tree of a valid composite witness, every node has at most k children.

Let $\mathcal{I}$ be the infinite rooted $(2d)$-ary tree just as in Section 4 of the paper. Let $\mathcal{I}_2$ be the infinite rooted binary tree (every node has 2 children). We say that a binary tree is *full* or *proper* if each of its nodes has either no or exactly two children. The following is a well-known fact that can be proved using Stirling's approximation for factorials:

Fact. For every $\epsilon > 0$, there are more than $(2-\epsilon)^{u-1}$ full binary subtrees of $\mathcal{I}_2$, for sufficiently large u.

c. Use the fact to prove that for every $\epsilon' > 0$, there are more than $((\sqrt{2} - \epsilon')2d)^{u-1}$ full binary subtrees of $\mathcal{I}$, for sufficiently large u and d.

d. Show that exploiting the property of valid composite witnesses found in Task (b) does not yield a substantial benefit in the analysis, that is even if we use this fact when counting in the proof of Lemma 4.1, the requirement $(\Gamma_F^+(C) \leq 2^{k-5})$ cannot be replaced by, say, $(\Gamma_F^+(C) \leq 2^{k-4})$.