Satisfiability of Boolean Formulas - Combinatorics and Algorithms
Autumn 2008

Exam

Remarks:

- The time might be too scarce to solve all the problems. Do not haste, not all points are required to achieve the maximum grade.
- The problems are not sorted, neither according to difficulty nor according to topic. Please read through all the task descriptions carefully and then decide on how you want to organise your work.
- Each of the 6 problems yields the same number of points.
- Remember to write your full name on each of the sheets you hand in.

Exam

Problem 1

Consider the class $\mathcal{L}$ of 3-CNF formulas F for which there exists a subset $W_0 \subseteq \text{vbl}(F)$ of size $|W_0| \leq \log(|F|)$ of the variables such that each clause $C \in F$ contains some variable from the set, i.e. $W_0 \cap \text{vbl}(C) \neq \emptyset$.

a. Let us call a set $F' \subseteq F$ of clauses *independent* if no two of them share a common variable, i.e. F' is an independent set in the dependency graph of F. Show that a formula in the class $\mathcal{L}$ cannot have an independent set of clauses of size larger than $\log(|F|)$.

b. Describe a polynomial-time algorithm that finds a subset $W \subseteq \text{vbl}(F)$ of size at most $3\log(|F|)$ such that each clause contains some variable from W. (Please note that the input consists only of F and that whilst the existence of a set W_0 as defined above is promised, W_0 is of course not part of the input and thus not known to your algorithm).

c. Describe a polynomial-time algorithm that for any formula $F \in \mathcal{L}$ decides whether F is satisfiable.

Problem 2

Let $V = \{x_1, x_2, \ldots, x_n\}$ and consider the following 3-CNF formula over V:

$$F := \{ \{x_{i_1}, \overline{x_{i_2}}, \overline{x_{i_3}}\} \mid 1 \le i_1 < i_2 < i_3 \le n \}.$$

a. Let $1 \le a < b < c \le n$ be integers. Give a resolution proof for the fact that $F \cup \{\{\overline{x_a}\}, \{x_b\}, \{x_c\}\}$ is unsatisfiable.

b. Determine the number $|\text{sat}_V(F)|$ of satisfying assignments.

c. Demonstrate that the algorithm ppz(...) outputs, on input F, a satisfying assignment for F with probability 1.

Problem 3

Let F be a 3-CNF over variable set V.

Suppose we sample assignments $\alpha \in \{0,1\}^V$ to the variables of F uniformly at random. Let X_F denote the random variable counting the number of violated clauses under α.

The *variance* of X_F is defined to be the average squared deviation of X_F from its mean, that is

$$\text{var}(X_F) := E((X_F - E(X_F))^2) = E(X_F^2) - E(X_F)^2$$

where the equality is derived by simply expanding the square and using linearity of expectation.

a. Give an example of a 3-CNF formula F for which the variance of X_F is zero.

b. Given two clauses $C, D \in F$, how can one compute the probability that both clauses are simultaneously violated?

c. Design a polynomial time algorithm that takes a 3-CNF formula F as input and computes the variance of X_F.

d. Design a polynomial time algorithm that takes a 3-CNF F as input and decides whether there exists an assignment that satisfies strictly more than 7/8 of the clauses of F. Your algorithm just has to output 'yes' or 'no'.

Problem 4

Let us call a CNF formula F *allowed* if every clause $C \in F$ is either a 2-clause consisting of two positive literals, e.g. $\{x, y\}$, or a 4-clause consisting of four negative literals, e.g. $\{\bar{x}, \bar{y}, \bar{z}, \bar{v}\}$.

Prove that every allowed CNF formula admits an assignment that satisfies at least $(1 - \Phi^4)$–fraction of the clauses, where $\Phi = \frac{\sqrt{5}-1}{2}$ is the golden ratio conjugate.

Problem 5

Let F be any 3-CNF formula. For any clause $D \in F$, let us define D^+ to be the set of positive literals in D. Let $D^+ = \{u_1, u_2, \ldots, u_l\}$. We say that D is *unsafe* if $l > 0$ and there exist l distinct negative clauses $C_1, C_2, \ldots, C_l \in \mathrm{Neg}(F)$ such that $\overline{u_i} \in C_i$ for $1 \leq i \leq l$. ($\mathrm{Neg}(F)$ is defined as in Chapter 7*).

Prove the following:

a. if there are <u>no</u> unsafe clauses, then F is satisfiable

b. if every negative clause $C \in \mathrm{Neg}(F)$ has a literal $\bar{x} \in C$ such that x is <u>not</u> contained in any unsafe clause, then F is satisfiable

Problem 6

Suppose F is a 3-CNF which admits a partial assignment $\alpha^* \in \{0,1\}^V$ for $V \subseteq \mathrm{vbl}(F)$ such that in every clause $C \in F$, α satisfies exactly one literal, dissatisfies exactly one literal and leaves the third literal unaffected.

Prove that for such a formula, Schöning's randomized $3 - SAT$ algorithm finds a satisfying assignment after an expected polynomial number of steps (here, we do not interrupt and restart the algorithm after $3n$ steps, we just let it run).