Exam "Satisfiability of Boolean Formulas"

December 21, 2007, 10:00 – 12:00, CAB G 52

Each of the six problems is granted the same number of points. You don't need to solve all of them: Completely solving five out of these problems will already give you the best grade. Don't forget to put your full name on the first page, and your initials on each of the other pages. You may use English and German, whatever you prefer.

Problem 0 — Encoding Satisfying Assignments

We consider the formula $\{\{x, \bar{y}\}, \{x, \bar{z}\}, \{y, \bar{u}\}, \{z, \bar{u}\}, \{u, \bar{x}, \bar{y}\}\}$ and the satisfying assignment α that sets all variables to 1.

0.1 Compute $\mathrm{enc}(\alpha, \pi_i)$ for $\pi_1 = (u, x, y, z)$, $\pi_2 = (u, y, z, x)$ and $\pi_3 = (x, y, z, u)$.

0.2 What is the expected length of $\mathrm{enc}(\alpha, \pi)$ for π drawn uniformly at random from the set of all permutations of $\{u, x, y, z\}$?

***

Problem 1 — Random Assignments

Let F be a CNF formula. Pick α uniformly at random from $\{0, 1\}^{\mathrm{vbl}(F)}$. Let Y_F denote the number clauses which are unsatisfied by α. Y_F is a random variable.
1.1 Consider $F = \{\{x, y\}, \{x, \bar{y}\}, \{\bar{x}, z\}, \{\bar{x}, \bar{z}\}, \{y, z\}, \{\bar{y}, \bar{z}\}\}$. What is $\mathbb{E}[Y_F]$? What is $\max(Y_F)$? What is $\min(Y_F)$? Give assignments achieving $\max(Y_F)$ and $\min(Y_F)$ and argue why this is really maximal / minimal.

1.2 Give a polynomial time algorithm that, given an input formula F, decides whether Y_F is a constant, i.e. whether all truth assignments satisfy an equal number of clauses.

***

Problem 2 — Signed Versions of Formulas

Consider the *unsigned* formula $F := \{\{x, y, z\}, \{x, y, u\}\}$. We obtain a *signed version of* F by replacing some variables in some clauses by their negation. For example, $\{\{\bar{x}, y, z\}, \{x, \bar{y}, u\}\}$ and $\{\{x, y, z\}, \{\bar{x}, \bar{y}, \bar{u}\}\}$ are signed versions of F.

2.1 Let $F_1 := \{\{x, y, z\}, \{x, y, u\}, \{x, y, v\}, \{y, z, u\}, \{y, z, v\}, \{y, u, v\}, \{z, u, v\}\}$. Show that every signed version of F_1 is satisfiable.

2.2 Let $F_2 := \{\{x, y, v\}, \{y, z, v\}, \{z, u, v\}, \{x, u, v\}, \{a, b, v\}, \{b, c, v\}, \{c, d, v\}, \{a, d, v\}\}$. Show that every signed version of F_2 is satisfiable.

***

Problem 3 — Flipping and Horn SAT

Flipping a variable x in a formula F means replacing each occurrence of x by $\bar{x}$ and vice versa. Flipping a set $U \subseteq \mathrm{vbl}(F)$ means flipping each variable in U. We denote the resulting formula by $\mathrm{flip}_U(F)$. For example, $\mathrm{flip}_{\{x,y\}}(\{\{x, y, z\}, \{\bar{x}, y, \bar{u}\}\}) = \{\{\bar{x}, \bar{y}, z\}, \{x, \bar{y}, \bar{u}\}\}$.

3.1 Let F be a CNF over V, and let $U \subseteq V$. Show that F is satisfiable if and only if $\mathrm{flip}_U(F)$ is satisfiable.

3.2 A *Horn formula* is a CNF formula in which each clause contains *at most one* positive literal. For example $\{\{x, \bar{y}\}, \{\bar{x}, \bar{y}, \bar{z}\}\}$ is a Horn formula, but $\{\{x, \bar{y}\}, \{x, y, \bar{z}\}\}$ is not. Show that a Horn formula F is unsatisfiable if and only if Unit Clause Reduction produces an empty clause.

3.3 We say a CNF formula F over variables V is *Horn flippable* if there is a set $U \subseteq V$ such that $\mathrm{flip}_U(F)$ is a Horn formula. Give a polynomial time algorithm deciding whether a given CNF formula is Horn flippable. Hint: Show how to construct a 2-CNF formula which is satisfiable if and only if F is Horn flippable.

***

Problem 4 — Hamiltonian Cycles and Paths in the Boolean Cube

4.1 Let $n \geq 1$ and let $x, y \in \{0, 1\}^n$ be two vertices of the boolean n-cube. Show that there exists a Hamiltonian path from x to y if and only if the Hamming distance $d_H(x, y)$ is odd.

4.2 Show that for $n \geq 2$, the boolean cube $\{0, 1\}^n$ has a Hamiltonian cycle.

***

Problem 5 — The Game of Life

The game of life is a popular simulation of a bacteria colony, where bacteria multiply, but also die of starvation if their local environment is too crowded. Formally, the game of life world is an $n \times n$ grid, and we describe its configuration as a bit matrix $A \in \{0, 1\}^{n \times n}$, where $a_{ij} = 1$ if a bacterium lives at field (i, j), and $a_{ij} = 0$ if that this field is empty.

The *neighborhood* of a field (i, j) consists of its 8 adjacent fields plus itself, formally the set $\{i-1, i, i+1\} \times \{j-1, j, j+1\}$. For simplicity, we add/subtract modulo n, i.e., the bacteria live in a "wrap-around world". Each configuration $A \in \{0,1\}^{n \times n}$ has a successor configuration $\delta(A) \in \{0,1\}^{n \times n}$, defined by the following rule: If the number of 1s in the neighborhood of (i, j) in matrix A is at least one and at most four, then $\delta(A)_{ij} = 1$, otherwise $\delta(A)_{ij} = 0$. If $\delta(A) = B$, then A is called the predecessor of B. It is not clear which configurations have a predecessor.

5.1 Give a high-level description how to construct formula from a given configuration $B \in \{0,1\}^{n \times n}$ that is satisfiable if and only if B has a predecessor.

5.2 How many variables does your formula have? What is the meaning of these variables? What size do your clauses have?