ETH Eidgenössische Technische Hochschule Zürich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Institute of Theoretical Computer Science
Update Friday, March 28
Emo Welzl, lecturer, and Timon Hertli, teaching assistant

Thursday, March 20, 2014

Satisfiability of Boolean Formulas *Spring 2014*
Special Assignment Set 1

- The solution is due on **Friday, April 4, 2014, 10:15 (strict!)**. Please bring a print-out of your solution with you to the exercise session. If you cannot attend, you may alternatively send your solution as a PDF to `timon.hertli@inf.ethz.ch`. We will send out a confirmation that we have received your file. Make sure you receive this confirmation within the day of the due date, otherwise complain timely.
- Please solve the exercises carefully and then write a nice and complete exposition of your solution using a computer, where we strongly recommend to use LaTeX. A tutorial can be found at `http://www.cadmo.ethz.ch/education/thesis/latex`.
- For geometric drawings that can easily be integrated into LaTeX documents, we recommend the drawing editor IPE, retrievable at `http://ipe7.sourceforge.net/` in source code and as an executable for Windows.
- You are welcome to discuss the tasks with your colleagues, but we expect each of you to hand in your own, individual write-up. **You should not share your write-up or (even worse) your source code!**
- There will be three special assignments this semester. Each of them will be graded and the average grade will contribute 30% to your final grade.
- This is a theory course, which means: if an exercise does not explicitly say "you do not need to prove your answer" or "justify intuitively", then a formal proof is **always** required.
- As with all exercises, the material covered in special assignments is relevant for the final exam.

Exercise 1 (SAT Algorithms) *(45 points)*

For a CNF formula F, define $m(F) := |F|$ to be the number of its clauses, and $l(F) := \sum_{C \in F} |C|$ to be the *length*, i.e. the total number of occurences in F.

In this exercise, we want to analyze randomized algorithms for satisfiability whose runtime depends on $m(F)$ and $l(F)$.

We allow the algorithms to give the wrong result with probability $\frac{1}{3}$. The intetion is to have algorithms with bounded running time. You are also allowed to give algorithms running in expected time but note that the expected running time needs to be bounded for all input formulas.

(a) Give a randomized algorithm deciding 3-SAT in expected time $O\left(\left(\frac{3}{2}\right)^{m(F)} \cdot \text{poly}(l(F))\right)$.

(b) Give a randomized algorithm deciding 3-SAT in expected time $O\left(\left(\frac{3}{2}\right)^{\frac{1}{3}l(F)} \cdot \text{poly}(l(F))\right)$.

(c) Give a randomized algorithm deciding k-SAT in expected time $O\left(\left(\frac{k}{2}\right)^{m(F)} \cdot \text{poly}(l(F))\right)$.

(d) Using the previous algorithm, build a randomized algorithm deciding SAT (of CNF formulas) in expected time $O\left(c^{l(F)} \cdot \text{poly}(l(F))\right)$ for constant c as small as you can achieve.
NOTE: You should try to make c as small as possible, but you don't have to prove any optimality.

(e) Give a randomized algorithm deciding SAT (of CNF formulas) in expected time $O\left(d^{m(F)} \cdot \text{poly}(l(F))\right)$ for some constant d.
NOTE: Any constant d will do, you don't have to be optimal here.

Exercise 2 (Simple Formulas) *(55 points)*

A *simple* CNF formula F is defined recursively as one of the following:

(i) $\{\square\}$ is simple.

(ii) If there is a variable x and simple formulas F_0 and F_1 where variable x does not occur, then $F = \bigcup_{C_0 \in F_0} \{\{\bar{x}\} \cup C_0\} \cup \bigcup_{C_1 \in F_1} \{\{x\} \cup C_1\}$ is simple.

We call a CNF formula F *insensitive* if every total assignment α satisfies the same number of clauses of F.

If a CNF formula F has no simple subformula, we call F *reduced*.

(a) Show that simple formulas are insensitive.

(b) Give a formula F that is not simple, insensitive, and every assignment satisfies all but one clause.

(c) Are all reduced formulas 2-satisfiable? 3-satisfiable?

(d) Show that an insensitive CNF formula F with $|F| \geq 2$ has clauses $C_1 \neq C_2$ with $\mathrm{vbl}(C_1) = \mathrm{vbl}(C_2)$.

(e) We say clauses C_1 and C_2 have a *simple resolvent* C if $C_1 = \{x\} \cup C$ and $C_2 = \{\bar{x}\} \cup C$; a simple resolution deduction is a resolution deduction where all obtained resolvents are simple.

Show that F has a simple resolution deduction of $\square$ if and only if F is not reduced.

(f) Suppose $F \neq \{\}$ is insensitive and every two different clauses $C_1, C_2 \in F$ conflict in *exactly* one variable (i.e. $|C_1 \cap \overline{C_2}| = 1$). Show that F is simple.

Challenge Problem (not graded): Can you decide in polynomial time whether a CNF formula is reduced?