Institute of Theoretical Computer Science
Emo Welzl, lecturer, and Timon Hertli, teaching assistant

Friday, May 10, 2013

Satisfiability of Boolean Formulas *Spring 2013*
Special Assignment Set 3

- The solution is due on **Friday, May 24, 2013, 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 (Many and Few Critical Clauses) *(60 points)*

In (a)-(d) we want to analyze how the *number* of critical clauses affects PPZ. In (e) and (f) we want to see what we can do when there are only few critical clauses.

In this exercise, we assume that F is a satisfiable 3-CNF formula over V. We fix a satisfying assignment α of F.

Remember: A clause C is *critical* for $x \in V$ (w.r.t. α) iff α sets exactly one literal of C to 1, and this literal is over x.

Definition 1. *Call $x \in V$* forced *if in* encode(α, F, V, π)*, at the step corresponding to x, we encounter $\{x\}$ or $\{\bar{x}\}$ (and hence we do* not *append $\alpha(x)$ to ϵ).*

Let $p_f(x)$ be the probability that x is forced *in* encode(α, F, V, π)*, for π chosen u.a.r. from all $n!$ permutations of V.*

Definition 2. *Let m_x be the number of clauses in F that are critical for $x \in V$ (w.r.t. α). If F is in 3-CNF, any such clause must be a 3-clause.*

In (a) and (b) (as in any other exercise, or in the exam), you don't need to reprove statements from the lecture notes. However you should clearly say what statements from the lecture notes you use, and you should argue why the statements can be used here.

(a) Argue that $\Pr_\pi(\mathrm{ppz}(F, V) = \alpha) \geq 2^{-\sum_{x \in V}(1 - p_f(x))}$.

(b) Argue that if $m_x = 0$, then $p_f(x) = 0$; and if $m_x = 1$, then $p_f(x) = \frac{1}{3}$.

(c) Suppose $m_x = 2$. What are the possible values for $p_f(x)$?

(d) Show that for arbitrarily large m_x, $p_f(x) \leq \frac{1}{2}$ is still possible (note that F is allowed to depend on m_x).

Challenge Problem (not graded): For fixed m_x, how do formulas look like that minimize or maximize $p_f(x)$?

(e) Let $m := \sum_{x \in V} m_x$. Give a polynomial time algorithm finding a satisfying assignment of F with probability at least $\left(\frac{2}{3}\right)^m$. Note that while m_x is defined w.r.t. a *fixed* satisfying assignment α, the algorithm only needs to find *some* satisfying assignment.

(f) Let $m := \sum_{x \in V} m_x$. Give a polynomial time algorithm finding a satisfying assignment of F with probability at least $\left(\frac{1}{2}\right)^{\frac{1}{2}|V|} \left(\frac{2}{3}\right)^{\frac{1}{8}m}$. Note that e.g. for $m = |V|$ this bound is better than the bound of (e).

Be careful that your analysis is formally correct!
HINT: You don't need to use PPZ. From the bound on the probability, the algorithm might be obvious. Use Jensen's inequality at some point.

Exercise 2 (Many Satisfying Assignments) *(40 Points)*

Remember that to analyze PPZ we use Lemma 5.4 to show that if the satisfying assignments are not very isolated, there are many of them (see top of p. 138 in the lecture notes). However, we estimated $2^{-(n-j(\alpha))/k} \geq 2^{-(n-j(\alpha))}$ which is not tight (except if there is a unique satisfying assignment). You might wonder whether we can do better if we are guaranteed to have many satisfying assignments. This is indeed the case. In this exercise we prove the following theorem:

Theorem 3. *If F is a $(\leq k)$-CNF over n variables with t satisfying assignments, then PPZ returns some satisfying assignment with probability $\left(\frac{t}{2^n}\right)^{1-\frac{1}{k}}$.*

The following lemma is crucial for the proof.

Lemma 4. *Let $S \in \{0,1\}^n$, nonempty. If s is chosen u.a.r. from S, then $|S| \geq 2^{E[\deg(s)]}$ where $\deg(s)$ is the degree of s in the subgraph of the n-cube induced by S.*

(a) Show that Lemma 4 implies Lemma 5.4 (lecture notes, p. 120). That is, prove Lemma 5.4 using Lemma 4 without using any properties of the n-cube.

(b) Prove Theorem 3 assuming Lemma 4.
HINT: In the analysis of PPZ, use Jensen's inequality again after summing over all satisfying assignments.

(c) Prove Lemma 4.

You may use the following inequality without proof: If $0 \leq q \leq r$, then

$$(q + r)\log_2(q + r) \geq q \log_2 q + r \log_2 r + 2q.$$

HINT: Prove that $|S| \log_2 |S| \geq \sum_{s \in S} \deg(s)$. Proceed by induction on n and partition the cube into two facets.