Special Exercise Set 3
November 24, 2009
due on Friday, November 27 in the exercise session
(same regulations apply as for the previous 2 sets)

3.1 Perfect Implementation

Let $f : \{0,1\}^V \mapsto \{0,1\}$ be a boolean function over the variable set V. Let F be a CNF formula over variable set V' with $V' \supseteq V$. We say F *perfectly implements* f if

$$\forall \alpha : V \mapsto \{0,1\} : \quad f(\alpha) = 1 \iff F^{[\alpha]} \text{ is satisfiable.}$$

In particular, if $V = V'$, then the assignments α for which $f(\alpha) = 1$ and are exactly the satisfying assignments of F.

Exercise a) Show that any k-clause (viewed as a boolean function) can be perfectly implemented by a 3-CNF formula.

Exercise b) Show that any boolean function can be perfectly implemented by some 3-CNF formula.

Exercise c) Show that if the function $x \vee y \vee z$ can be perfectly implemented by a 2-CNF formula, then $\mathcal{P} = \mathcal{NP}$.

Exercise d) Let $\alpha_1, \alpha_2, \alpha_3 : \{0,1\}^V$ be truth assignments. We define the truth assignment $\text{maj}(\alpha_1, \alpha_2, \alpha_3)$ as follows:

$$\text{maj}(\alpha_1, \alpha_2, \alpha_3)(x) = \begin{cases} 1 & \text{if at least two of } \alpha_1, \alpha_2, \alpha_3 \text{ set } x \text{ to } 1\ , \\ 0 & \text{else.} \end{cases}$$

Let F be a 2-CNF formula. Show that if $\alpha_1, \alpha_2, \alpha_3$ all satisfy F, then $\text{maj}(\alpha_1, \alpha_2, \alpha_3)$ also satisfies F.

Exercise e) Show that $x \vee y \vee z$ cannot be implemented by a 2-CNF formula.

3.2 Bounded Degree SAT

Let F be a CNF formula. The degree of a variable x in F, denoted by $\deg_F(x)$, is the number of clauses of F containing x or $\bar{x}$. A degree-d CNF formula is a CNF formula F where $\deg_F(x) \leq d$ for any variable x. Degree-d SAT is the problem of deciding whether a given degree-d CNF formula is satisfiable. Formally, Degree-d SAT is the language consisting of all satisfiable degree-d CNF formulas.

Exercise a) Show that Degree-2 SAT is in P.

Exercise b). Show that Degree-3 SAT is NP-complete. That is, give a polynomial time algorithm that takes a CNF formula F as input and outputs a sat-equivalent degree-3 CNF formula F'. You might use the following degree-2 CNF formula:

$$\{\bar{x}_1, x_2\}, \{\bar{x}_2, x_3\}, \ldots, \{\bar{x}_{n-1}, x_n\}, \{\bar{x}_n, x_1\} ,$$

which is satisfied if and only if all variables $x_1, \ldots, x_n$ are set to the same value.

Now we consider degree-d k-CNF formulas, that is, formulas F where $|C| = k$ for all $C \in F$ and $\deg_F(x) \leq d$ for all $x \in \mathrm{vbl}(F)$. Again, Degree-d k-SAT is the language of all satisfiable degree-d k-CNF formulas.

Exercise c) Show that for any $k \geq 1$, Degree-$\frac{2^{k-2}}{k}$ k-SAT is in P.

Exercise d) Suppose your friend gives you, as a birthday present, an unsatisfiable degree-d k-CNF formula with $k \geq 3$. Using this formula, give a polynomial algorithm that takes as input a degree-3 CNF formula F and outputs a degree-d k-CNF formula F' such that F' is satisfiable if and only if F is. Please note that in F', every clause must have size exactly k, and also that it is not your choice what d and k are.

Bonus Exercise e) Prove that for all $d, k \geq 0$, at least one of the following holds: (i) Degree-d k-SAT is in P, or (ii) Degree-d k-SAT is NP-complete.

3.3 Algorithmic Local Lemma

Exercise a) *Incompressibility* plays a role in the constructive proof of the Lovász Local Lemma. Here we want to investigate it in a somewhat simpler scenario. Consider a function $f : \{0,1\}^n \mapsto \{0,1\}^*$ that is *prefix free*, i.e. there do not exist $x_1, x_2 \in \{0,1\}^n$, $x_1 \neq x_2$ and $y \in \{0,1\}^*$ such that $f(x_1) = f(x_2)y$ (where $f(x_2)y$ is simply the concatenation of two strings). Let $|f(x)|$ denote the length of the string $f(x)$. Show that

$$\mathbb{E}_{x \in_{\text{u.a.r.}} \{0,1\}^n}[|f(x)|] \geq n ,$$

where $x \in_{\text{u.a.r.}}$ means that x is sampled uniformly at random from $\{0,1\}^n$. In other words, a random string from $\{0,1\}^n$ cannot be compressed. You might want to use Lemma 5.3 from the lecture notes together with the following lemma:

Lemma 1.1 (Special case of Jensen's inequality) *Let X be a random variable. Then $\mathbb{E}[2^X] \geq 2^{\mathbb{E}[X]}$.*

Exercise b) The algorithm `solve_lll` outputs a log consisting of at most $m\lceil \log m \rceil + (k-1)t$ bits. Modify the algorithm, i.e., modify the format of the log it produces such that the log consists of at most $m + (k-1)t$ bits. Hint: in `solve_lll`, we write the binary code for every clause C we are treating. Show how to encode the same information with only *one* bit per clause!