Institute of Theoretical Computer Science Spring semester, 2015
Emo Welzl, lecturer, and Jerri Nummenpalo, teaching assistant

Satisfiability of Boolean Formulas Spring 2015
Master Solutions, Special Assignment Set 1

Exercise 1 (Weighed 2-satisfiable formulas) *(40 points)*

Let us use notation similar to the one used in [1] in the proof of Theorem 3.4 there. Let $V = \text{vbl}(F)$. Define the sets

$$F^+ = \{C \in F \mid C \text{ has exactly one positive literal}\},$$
$$F^- = \{C \in F \mid C \text{ has no positive literals}\}$$

and for all $x \in V$ define

$$d^+(x) = \sum_{C \in F^+,\, x \in C} \mu(C)$$
$$d^-(x) = \sum_{C \in F^-,\, \overline{x} \in C} \mu(C)$$

Now switch variables in F (i.e. switch x to $\overline{x}$ and vice versa and then the sets F^+ and F^- change) until the following conditions are met

(i) $d^+(x) \geq d^-(x) \quad \forall x \in V$,

(ii) $d^+(x) \geq d^-(x) + 1, \quad \text{if } \{\overline{x}\} \in F$.

Note that we can only guarantee to satisfy condition (ii) if the weights take integer values. We then have that

$$\begin{aligned}
\mu(F^+) = \sum_{x \in V} d^+(x) &= \sum_{\substack{x \in V, \\ \{\overline{x}\} \in F}} d^+(x) + \sum_{\substack{x \in V, \\ \{\overline{x}\} \notin F}} d^+(x) \\
&\geq \sum_{\substack{x \in V, \\ \{\overline{x}\} \in F}} (d^-(x) + 1) + \sum_{\substack{x \in V, \\ \{\overline{x}\} \notin F}} d^-(x) \\
&= \sum_{\substack{x \in V, \\ \{\overline{x}\} \in F}} 1 + \sum_{x \in V} d^-(x) = \sum_{\substack{x \in V, \\ \{\overline{x}\} \in F}} 1 + \sum_{C \in F^-} |C| \cdot \mu(C) \\
&= \sum_{\substack{C \in F^-, \\ |C| = 1}} (1 + \mu(C)) + \sum_{\substack{C \in F^-, \\ |C| \geq 2}} |C| \cdot \mu(C) \qquad (1)
\end{aligned}$$

Now, if the formula in (1) is larger than $2\mu(F^-)$ then for the all-one assignment we have that the only clauses not satisfied are in $|F^-|$ and combining the inequalities

$$\mu(F^+) + \mu(F^-) \leq \mu(F)$$
$$\mu(F^+) \geq 2\mu(F^-)$$

gives us that $\mu(F^-) \leq \frac{\mu(F)}{3}$ so at least $\frac{2}{3}$ of the total weight is satisfied. Now to the actual questions.

(a) $W = [0, 1]$. Looking at discussion following the proof of Theorem 2.4 in which the tightness of the result is shown, we see that the same idea holds in this case as we can always scale the weights.

(b) Let $W = \{1, 2, \ldots, k\}$ for some $k \in \mathbb{N}_{>0}$. Now we restrict the 1-clauses to have weight 1. Then from formula (1) we see that that $\mu(F^+) \geq 2\mu(F^-)$ and we're done.

(c) Let $W = [1, k]$. In this case we cannot guarantee the condition (ii) to hold. Consider for example a formula where the only occurrence of x is in two clauses such as $\{x\}$ and $\{\overline{x}, \overline{y}\}$ with weights $1 + \epsilon$ and 1 for some $\epsilon > 0$ respectively. Then we only have that $d^+(x) = d^-(x) + \epsilon$.

(Bonus 1) As some people observed, the formula

$$F = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{\overline{x}_1, \overline{x}_2\}, \{\overline{x}_1, \overline{x}_3\}, \{\overline{x}_1, \overline{x}_4\}, \{\overline{x}_2, \overline{x}_3\}, \{\overline{x}_2, \overline{x}_4\}, \{\overline{x}_3, \overline{x}_4\}\}$$

with weights 2 on the 1-clauses and weights 1 on the 2-clauses has the property that only a $\frac{9}{14}$ fraction of the weight is satisfiable. To see this, note that the weight satisfied depends only on the number of variables set to true and then consider what is the maximum weight one can satisfy. This works as an example because $\frac{9}{14} = \frac{27}{42} < \frac{28}{42} = \frac{2}{3}$.

Exercise 2 (NAE-Satisfiability) *(20 points)*

(a) For example $\{\{x, y\}, \{x, \overline{y}\}\}$. The assignments $(1, 1), (0, 0)$ violate the NAE-satisfiability property with the first clause and the assignments $(1, 0), (0, 1)$ violate it with the second clause.

(b) For example $\{\{x, y, z\}, \{\overline{x}, y, z\}, \{x, \overline{y}, z\}, \{x, y, \overline{z}\}\}$. Each clause violates exactly two of the total of eight assignments and these assignments are distinct so there is no assignment satisfying the NAE condition.

(c) For any clause exactly two assignments are not NAE-satisfiable and the rest are. If there are less than 2^{k-1} clauses in the formula, then by a simple counting argument there has to exist one that is NAE satisfiable for all of the clauses. One can also in probabilistic terms.

(d) We generalize the example from (a) and (b). Group all the 2^n assignments into 2^{n-1} pairs where each pair consists of an assignment and the negation of the assignment where every bits gets a different value. Then, for each such pair add the unique clause to the formula which is not NAE-satisfiable for that pair.

Exercise 3 (Only double conflicts) *(20 points)*

(a) If for all $C, D \in F$ we have that $|C \cap \overline{D}| \neq 1$ then say that F satisfies the *complementarity condition*. We will prove the claim that F is satisfiable by induction on the number of clauses in F. If there is just one clause in F then F satisfies the complementarity condition. Now assume that all formulas of size less than m satisfying the complementarity condition are satisfiable. Let F be a formula of size m, $C \in F$ a clause, and $u \in C$ a literal. Additionally, assume that F satisfies the complementarity condition. Partition the clauses in F into three classes: $F_u, F_{\overline{u}}$ and F_0 which are those that contain the literal u, those that contain the literal $\overline{u}$ and those that contain neither, respectively.

Consider the formula $F' := F^{[u \mapsto 1]}$. Note that $F' = F_{\overline{u}}^{[u \mapsto 1]} \cup F_0$ as all the clauses with u are satisfied. None of the clauses in F' is the empty clause due to the complementarity assumption on F. We need to show that F' also satisfies the complementarity condition and because $|F'| < m$ we are then done by induction. Towards that end, letting $D, E \in F'$ we want to show that $|D \cap \overline{E}| \neq 1$. If D and E are both from F_0 the statement is clear as $D, E \in F$. The same is true if they are both from $F_{\overline{u}}^{[u \mapsto 1]}$ because we only removed the same literal from both of them. If one of them is from $F_{\overline{u}}^{[u \mapsto 1]}$ and one is from F_0, the statement still holds, as F_0 does not contain $\overline{u}$. Therefore F' satisfies the complementarity condition and because $|F'| < m$ we are done by induction.

(b) The previous statement suggests a straightforward algorithm: Pick any clause in F and some literal u in it. Set u so that the clause is satisfied. Now all clauses in F_u are satisfied and we are left with a smaller formula. Repeat as long as there are clauses in the formula. No clause that is already satisfied can become unsatisfied.

References

[1] Claudia Käppeli and Dominik Scheder. Partial satisfaction of k-satisfiable formulas. *Electronic Notes in Discrete Mathematics*, 29:497–501, 2007.