Institute of Theoretical Computer Science Tuesday, May 5, 2015
Emo Welzl, lecturer, and Jerri Nummenpalo, teaching assistant

Satisfiability of Boolean Formulas Spring 2015
Master Solutions, Special Assignment Set 2

Exercise 1 (Monotone CNF complexity) *(50 points)*

a) We will first show that $c(f) \leq |\textsc{MaxUnsat}(f)|$ by exhibiting a CNF formula F which is equivalent to f and has the appropriate size. For an assignment $\alpha \in \{0,1\}^n$ and for a monotone boolean function f let

$$C_\alpha = \{x_i \mid \alpha(i) = 0\}, \text{ and}$$
$$F = \{C_\alpha \mid \alpha \in \textsc{MaxUnsat}(f)\}.$$

Now we claim that $F \equiv f$. To see this, let $\alpha \in \{0,1\}^n$ be an assignment. If $f(\alpha) = 0$, then there exists an assignment $\beta \in \{0,1\}^n$ satisfying $\alpha \leq \beta$ and $\beta \in \textsc{MaxUnsat}(f)$ which implies that $C_\beta \in F$. As $C_\beta^{[\beta]} = \Box$, $\alpha \leq \beta$ and C_β consists only of positive literals we also have that $C_\beta^{[\alpha]} = \Box$ and therefore $F(\alpha) = 0$. On the other hand, if $f(\alpha) = 1$ then assume on the contrary that $F(\alpha) = 0$. Then there exists an assignment β so that $C_\beta \in F$ and $C_\beta^{[\alpha]} = \Box$. But then necessarily $\alpha \leq \beta$ because α has to agree with β on the valuse in the clause and β has all its other entries as 1. As $\beta \in \textsc{MaxUnsat}(f)$, it implies that $f(\beta) = 0$ and from the monotonicity of f we then have that also $f(\alpha) = 0$, a contradiction.

Therefore indeed $F \equiv f$. We have that $|F| = |\textsc{MaxUnsat}(f)|$ and F is a (monotone) CNF which establishes $c(f) \leq |\textsc{MaxUnsat}(f)|$.

Towards showing $c(f) \geq |\textsc{MaxUnsat}(f)|$ let F be a CNF formula equivalent to f. For any assignment $\alpha \in \textsc{MaxUnsat}(f)$ the formula F has a clause which is not satisfied by α. Denote such a clause by D_α. We will show that for distinct $\alpha, \beta \in \textsc{MaxUnsat}(f)$ we also have $D_\alpha \neq D_\beta$ which establishes the result. Assume on the contrary that $D_\alpha = D_\beta$. Then define a new assignment γ which agrees with α and β on the variables appearing as literals in D_α. For the rest of the values, choose the component-wise maximum of α and β. We have $\alpha < \gamma$ and $\beta < \gamma$ because $\alpha \neq \beta$ and both α and β are in $\textsc{MaxUnsat}(f)$ which implies that they are incomparable. But now also $F(\gamma) = 0$ which is a contradiction to α and β being maximally unsatisfying.

b) Let f monotone boolean function. The direction $c(f) \leq mc(f)$ is clear. For the other direction let F be a CNF formula so that $|F| = c(f)$ and $F \equiv f$. Let F' be the CNF formula attained from F by removing all the negative literals from all the clauses. We claim that $F' \equiv F$ which establishes $mc(f) \leq c(f)$. To see this let $\alpha \in \{0,1\}^n$ be an assignment. If $F(\alpha) = 0$ then also $F'(\alpha) = 0$. If $F(\alpha) = 1$ and we had contrary to the claim that $F'(\alpha) = 0$ then there exists a clause $C \in F$ such that C is satisfied under α, but removing the negative literals from C makes it unsatisfiable under α. Let then β be the assignment attained from α by setting all the values corresponding to the removed literals to 1 and otherwise setting them the same as in α. That is, if the literal $\overline{x}_i$ was removed then set $\beta(i) = 1$. Now $\alpha \leq \beta$, but F is not satisfiable under β which is a contradiction to monotonicity. This establishes that $mc(f) \leq c(f)$.

Note that we could have also relied on the a)-part as the same proof goes through when we consider only monotone CNF formulas.

c) There are 2^n maximally unsatisfying assignments (for each DNF-clause pick one of the two variables to be true and one of them to be false) which via the help of the previous results proves the claim as the formula is monotone. An assignment satisfies the formula iff it assigns 1 to both x_{j1} and x_{j2} for at least one $j \in \{1, \ldots, n\}$. Therefore an equivalent CNF formula matching this bound is

$$\bigwedge_{j_i \in \{1,2\}, i=1,2,\ldots,n} (x_{1j_1} \vee x_{2j_2} \vee \ldots \vee x_{nj_n})$$

which contains 2^n clauses.

d) Think of a monotone boolean function f as the Boolean lattice over the set $[n]$ where elements are labeled with 1 or 0 according to the function value. Then this function is specified the elements in

$\textsc{MaxUnsat}(f)$ which are all pairwise incomparable i.e. they form an antichain. Therefore the question is what is the maximal cardinality of an antichain in the Boolean lattice? The answer is given by Sperner's Theorem [1] which states that such an antichain always has size at most $\binom{n}{\lfloor n/2 \rfloor}$. But choosing all the bit vectors with $\lfloor n/2 \rfloor$ ones gives us an antichain whose cardinality is exactly that so the bound is tight. So the function in question would map all bitvectors having $\lfloor n/2 \rfloor$ or more ones into 1 and the rest to 0.

Exercise 2 (Neighborhood graph) *(50 points)*

a) An equivalent question is to ask how the graph determines the maximum number of clauses that are not satisfied by a total assignment. An independent set in a graph is a subset I of the vertices such that no two vertices in I share an edge. A maximal independent set is an independent set which is not a proper subset of any other independent set. Note that for any independent set in the simple neighborhood graph of some formula F we can find a total assignment which satisfies none of the clauses in the independent set. On the other hand, for any assignment α, the set of clauses not satisfied by α is an independent set in the simple neighborhood graph. If it wasn't, there would be a pair of unsatisfied clauses sharing an edge which means they would share a pair of complementary literal: a contradiction to the unsatisfiability of the clauses.

Therefore the maximum number of clauses not satisfied by a total assignment is determined by the maximum cardinality independent set. Let $I_{\max}$ be such a maximum size independent set of the neighborhood graph of some CNF formula F. Then the minimum number of clauses satisfied in F by a total assignment is then $|F| - |I_{\max}|$. This is the same as the size of the minimal vertex cover.

b) Note that for any $C, D \in F$ we have that $|C \cap \overline{D}| \neq 1$. By Special assignment 1, Exercise 3 we know that F is satisfiable.

c) For a given multigraph $M = (V, E, w)$ construct a CNF formula as follows. Start with $|V|$ clauses which are initially empty which we identify with the vertices of M. For every $e = (C, D) \in E$ add $w(e)$ many unique variables to C and add their negations to D. If there are clauses remaining which are empty (corresponding to vertices with no adjacent edges) add a new variable for each of them which makes them 1-clauses.

d) Let the variables of F be $\{x_1, x_2, \ldots, x_n\}$ and let $M = (F, E, w)$ be the neighborhood graph with multiplicities of F. For every $i = 1, \ldots, n$, define the graphs $K(x_i) = (V(x_i), E(x_i))$ by $V(x_i) = \{C \in F \mid x_i \in C \text{ or } \overline{x}_i \in C\}$ and $E(x_i) = \{(C, D) \in E \mid E \subseteq V(x_i)\}$. Considering M as a multigraph we can write $M = \bigsqcup_{i=1}^{n} K(x_i)$. Note that each $K(x_i)$ is a complete bipartite graph and every (multi)-edge e in M is present in precisely $w(e)$ graphs $K(x_i)$.

e) Let us first deal with the general case.

General CNF formula

With every clause associate an assignment $\alpha \in \{0, 1\}^n$ which will be its *color* according to the following rule: For every clause choose an assignment which doesn't satisfy the clause. If there is ambiguity (i.e. if the clause is not an n-clause) then pick any of the possible assignments. We claim that this is a proper coloring of the simple neighborhood graph. To show that, assume on the contrary that there existed two clauses with the same assignment α associated to them. Then α doesn't satisfy either of these two clauses, but as they share an edge they must have a complementary literal which is a contradiction to being unsatisfied under the assignment α. We used 2^n colors which proves the claim. Note that 2^n is also a lower bound on the chromatic number if we consider all the n-clauses as their simple neighborhood graph is a 2^n-clique.

k-CNF formula

We consider a series of improving bounds for the case of a k-CNF formula.

Upper bound 1. Let F be the k-CNF formula over n variables which contains all the possible k-clauses of which there are $\binom{n}{k}2^{n-k}$ many. Denote by G the simple neighborhood graph of F. The graph G has the largest chromatic number over all simple neighborhood graphs of k-CNF formulas as any other simple neighborhood graph is an induced subgraph of it. By Brooke's theorem the maximum degree of a graph gives an upper bound on the chromatic number (as long as our graphs is not the complete graph or an odd cycle in which case we would have to add 1 to the maximum degree). This is often a good first approach in estimating the chromatic number. All the vertices in G have the same degree and we can estimate the maximum degree as

$$\Delta(G) \leq \sum_{i=1}^{k} \binom{k}{i} \binom{n-i}{k-i} 2^{k-i}$$

where we sum over the number of complementary variables in the neighbors of a vertex of G and for each i complementary literals there are $\binom{k}{i}$ choices of which to pick. Then, for the remaining variables there are $\binom{n-i}{k-i}$ choices which we can have as positive or negative literals, hence the term 2^{k-i}. Note that this overcounts the number of edges, but this formula is easier to analyze than the exact degree, and this doesn't give us the best bound anyways. Namely, using the bound $\binom{n-i}{k-i} \leq \binom{n-1}{k-1}$ we get that

$$\Delta(G) \leq \binom{n-1}{k-1} \sum_{i=1}^{k} \binom{k}{i} 2^{k-i} = \binom{n-1}{k-1} (3^k - 2^k).$$

Therefore we have established that $\chi(G) \leq \binom{n-1}{k-1}(3^k - 2^k)$.

Upper bound 2. Another approach at an upper bound is to consider the following actual coloring which resembles the one we used in the beginning for the general case. Given a k-CNF formula F and its simple neighborhood graph G, associate with each clause C an assignment $\alpha \in \{0,1\}^n$ as its color so that all literals in C are mapped to 1 under α and the variables not appearing in C are mapped to 1 under α. If the clause consists only of negative literals then abbreviate from this rule and color the clause using the all 0's assignment. Now any two clauses sharing a complementary literal (having an edge in the neighborhood graph) will have a different assignment as its color. Additionally, we only used assignments with at most $k-1$ 0's in them and the all 0 assignment for the negative clauses. This gives an upper bound on the chromatic number:

$$\chi(G) \leq 1 + \sum_{i=0}^{k-1} \binom{n}{i} = 1 + \mathrm{vol}(n, k-1)$$

where the sum counts the number of zeros in the assignments we used as colors and the additional 1 is for the all zeroes assignment.

Upper bound 3. Further improving our bound, let C again be a k-clause in the k-CNF F. If C has less or equal to $k/2$ negative literals, then color C as in the previous bound so that C gets associated the assignment which is the all 1's vector except for the places corresponding to negative literals in C. On the other hand if C has more than $k/2$ negative literals then associate to C the assignment which is all 0 except at the places which correspond to the positive literals of C which get a 1. Now again two clauses sharing a complementary literal will have a different assignment associated to them. To see this, consider the case when the clauses both come from the same 'coloring scheme' i.e. they both have a majority of negative literals or they both have a majority of positive literals. Then the same argument applies as before. If they come from a different coloring scheme, then the number of 1's and 0's in the assignment is different (assuming $k \neq n$ in which case we already have established a tight bound of 2^n) which means the assignments can't be the same. Taking the parity into account this gives us a bound of

$$\chi(G) \leq \begin{cases} 2 \cdot \mathrm{vol}(n, k/2) & \text{if } k \text{ is even,} \\ \binom{n}{(k+1)/2} + 2 \cdot \mathrm{vol}(n, k/2) & \text{if } k \text{ is odd.} \end{cases}$$

Upper bound 4. Consider a coloring of the simple neighborhood graph of the formula over n variables containing all the k-clauses. For every color class we can associate an n-clause so that all the k-clauses having that color are subsets of the set of those n literals appearing in the n-clause. Therefore, the question of finding a minimal coloring is equivalent to finding a minimal collection $\mathcal{F}$ of n-clauses such that every k-clause is a subset of some n-clause in $\mathcal{F}$.

We can build such a set randomly by picking n-clauses u.a.r. from the set of all n-clauses. Then a fixed k-clause is covered by such a random n-clause with probability $\left(\frac{1}{2}\right)^k$. After picking r such n-clauses the expected number of k-clauses not covered is

$$\mathbb{E}[\text{The number of } k\text{-clauses not covered}] = \binom{n}{k} 2^k \left(1 - \left(\frac{1}{2}\right)^k\right)^r.$$

That quantity is less than 1 when

$$r > \frac{1 - k \ln 2 - \ln \binom{n}{k}}{\ln \left(1 - \left(\frac{1}{2}\right)^k\right)} > 2^k \left(k \ln 2 + \ln \binom{n}{k} - 1\right) > 2^k \left(k \ln 2 + n \ln 2 - 1\right) = \mathcal{O}(2^k(n+k)) = \mathcal{O}(2^k n).$$

Upper bound summary The last upper bound we established is the best one we attained. It remains open still to construct such a coloring. Here we see again the power of probabilistic approach.

Lower bounds For a lower bound we note that considering all the k-clauses over a fixed set of k variables gives a lower bound of 2^k. Therefore there is still a factor of n difference between the lower bound and the upper bound.

f) Consider for example the formula

$$\{\{x, y, z\}, \{\overline{x}, y, z\}, \{a, b, \overline{z}\}\}$$

which is satisfiable, but has no multiple edges in the neighborhood graph with multiple edges.

g) As the simple neighborhood graph of F is complete, every pair of clauses shares a pair of complementary literals. Because F is also unsatisfiable, by Exercise 2.2. we have that $\sum_{C \in F} 2^{-|C|} = 1$.

References

[1] Emanuel Sperner. Ein satz über untermengen einer endlichen menge. *Mathematische Zeitschrift*, 27 (1):544–548, 1928.