Problem Set no. 1

Given: Nov 2, 2020
Due: Nov 18, 2020

Exercise 1.1 Prove that the CDCL solver described in class with nonchronological backtracking never makes more than $O^*(2^n)$ decisions and implications. (This dominates also the number of clauses it learns because we can map each new clause to the new implication of the UIP right after we learn it.) The O^* notation hides polynomial factors.

Exercise 1.2 Prove by induction on the number of variables that if a CNF formula is unsatisfiable then we can derive the empty clause by a sequence of resolution steps. (hint: consider $F[x = 0]$ and $F[x = 1]$ for some arbitrary variable x.)

Exercise 1.3 Consider the CNF formula $(\overline{x_1} \vee x_2) \wedge (\overline{x_3} \vee x_4) \wedge \cdots \wedge (\overline{x_{2n-1}} \vee x_{2n}) \wedge (w \vee y) \wedge (y \vee \overline{w}) \wedge (z \vee \overline{y}) \wedge (\overline{y} \vee \overline{z})$.
a) Assume we run DPLL with unit propagation but without CDCL. Assume also that when we have to take a decision we pick the first unassigned variable in the list $x_1, x_2, x_3, \ldots, x_{2n-1}, x_{2n}, w, y, z$ and give it first the value 1. Describe shortly the way the algorithm operates, what would be the running time as a function of n? justify your answer.
b) Assume now we run the algorithm with CDCL and nonchronological backtracking as described in class. Describe the run of the algorithm, which clauses are learnt and from which conflict graphs? What is the running time as a function of n ?
c) Prove the best bound you can on the running time of DPLL with CDCL and nonchronological backtracking on (arbitrary) 2-SAT formula. Prove your answer.

Exercise 1.4 Sketch an SMT solver that finds a satisfying assignment of a propositional formula over predicates of the form $x \neq y$ and $x = y$ or decides that no such assignment exists.

Exercise 1.5 Prove that the success probably of Schöning algorithm for an instance of k-SAT is $1/(2 - 2/k)^n$. Repeat the steps of the proof given in class for 3-SAT and indicate what are the required modifications.

Exercise 1.6 Consider the following algorithm for 2-SAT. Start with an arbitrary assignment. Repeat the following step $2n^2$ times: Pick an arbitrary unsatisfied clause, pick uniformly at random one of the literals of the clause and flip the value of the corresponding variable, terminate if the formula is satisfiable. If the loop has not terminated before running through it $2n^2$ times then assert that the formula is not satisfiable.
a) Prove that if the formula is satisfiable then the expected number of steps until we find a satisfying assignment is at most n^2. (Hint: Consider a random walk on the integers $0, 1, \ldots, n$. From n we go to $n - 1$ with probability 1. When at 0 we stay at 0. When we are at j we go to $j + 1$ with probability $1/2$ and to $j - 1$ with probability $1/2$. Define h_j to be the expected number of steps to reach 0 when we start at j. Then $h_0 = 0$, $h_n = h_{n-1} + 1$, and $h_j = (h_{j-1} + 1)/2 + (h_{j+1} + 1)/2$. Find h_j, formulate the connection between this walk and the algorithm.)
b) Use Markov inequality to show that if the formula is satisfiable we find a satisfying assignment with probability at least $1/2$.
c) How would you change the algorithm so that the probability that we mistakenly report that a satisfiable formula is unsatisfiable with probability at most $1/2^k$ for some fixed $k > 1$.