

## Homework Assignment 2

### Due Tuesday, February 13

The answers to the homework assignment should be your own individual work. Please hand in a hard copy of your solutions in class on the due date.

1. (15 points, 5 points each) A solution to a **graph coloring** problem is an assignment of colors to vertices such that no two adjacent vertices have the same color. Formally, a finite graph  $G = \langle V, E \rangle$  consists of vertices  $V = \{v_1, \dots, v_n\}$  and edges  $E = \{\langle v_{i1}, w_{i1} \rangle, \dots, \langle v_{ik}, w_{ik} \rangle\}$ . The finite set of colors is given by  $C = \{c_1, \dots, c_m\}$ . A problem instance is given by a graph and a set of colors: the problem is to assign each vertex  $v \in V$  a color  $\text{color}(v) \in C$  such that for every edge  $\langle v, w \rangle \in E$ ,  $\text{color}(v) \neq \text{color}(w)$ . Clearly, not all instances have solutions.

Show how to encode an instance of a graph coloring problem into a PL formula  $F$ .  $F$  should be satisfiable iff a graph coloring exists.

- (a) Describe a set of constraints in PL asserting that every vertex is colored. Since the sets of vertices, edges, and colors are all finite, use notation such as “ $\text{color}(v) = c$ ” to indicate that vertex  $v$  has color  $c$ . Realize that such an assertion is encodeable as a single propositional variable  $P_v^c$ .
  - (b) Describe a set of constraints in PL asserting that every vertex has at most one color.
  - (c) Describe a set of constraints in PL asserting that no two connected vertices have the same color.
2. (15 points) Describe the execution of the DPLL algorithm (using BCP and PLP but without learning) on the following formula.

$$(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r) \wedge (\neg p \vee q \vee r) \wedge (\neg q \vee r) \wedge (q \vee \neg r)$$

3. (20 points) Consider the following set of clauses:

$$\begin{aligned} c_1 : & (x_5 \vee \neg x_1 \vee x_3) \\ c_2 : & (\neg x_1 \vee \neg x_5) \\ c_3 : & (\neg x_3 \vee \neg x_4) \\ c_4 : & (x_1 \vee x_4) \\ c_5 : & (x_1 \vee \neg x_4) \\ c_6 : & (\neg x_1 \vee x_5) \end{aligned}$$

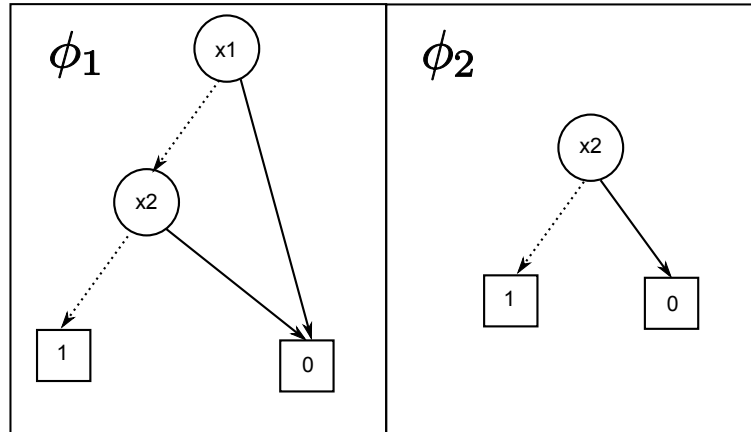
Show how a CDCL-based SAT solver decides the satisfiability of these clauses. For choosing the next assignment in the Decide step, use the dynamic largest individual sum (DLIS) heuristic, and in the case of a tie, favor variable  $x_i$  with the lowest index  $i$ . If there is a tie between  $x_i$  and  $\neg x_i$ , pick  $x_i$ . For deriving conflict clauses, use the latter strategy we discussed in class using first unique implication points, and backtrack to the second highest decision level in the derived conflict clause. Show the implication graph at each decision level, and keep all conflict clauses derived in AnalyzeConflict.

4. (15 points) Show how to construct the BDD for the formula

$$\neg(x_1 \vee (x_2 \wedge \neg x_3))$$

using variable order  $x_1, x_2, x_3$ , and starting with an ordered binary decision tree.

5. (20 points) Consider the BDDs for formulas  $\phi_1$  and  $\phi_2$  shown below:



- (a) (3 points) Show the BDD for  $\neg\phi_1$ .
- (b) (3 points) Show the BDD for  $\neg\phi_1 \downarrow x_1$
- (c) (3 points) Show the BDD for  $\neg\phi_1 \downarrow \neg x_1$
- (d) (11 points) Show how to construct the BDD for  $\neg\phi_1 \wedge \phi_2$  using the Apply procedure and using your BDDs from parts (a)-(c). Use the variable order  $x_1 < x_2$ . Your solution must clearly show the final BDD representing  $\neg\phi_1 \wedge \phi_2$ .