Verification of Digital Systems, Spring 2019

J. A. Abraham

Homework No. 1

DUE: February 14, 2019

Total Points: *150*

1. Assume we have the following DNF formula:

$$x_1 + x_2.x_3 + x_4.x_5.x_6$$

where x_i terms (i from 1 to 6) are different variables.

(a) (5 points) What is the corresponding CNF form for the above formula? How many clauses does it have?

(b) (15 Points) How many clauses will the CNF formula have if we have a general form of the above DNF formula with n clauses and $\frac{n(n+1)}{2}$ variables, which is:

$$x_1 + x_2.x_3 + ... + x_{\frac{n(n-1)}{2}+1}.x_{\frac{n(n-1)}{2}+2}...x_{\frac{n(n+1)}{2}}$$

**You are not allowed to define new variables.*

2. Use the basic SAT algorithm to find an assignment which satisfies F=1 in each of the following formulas:

(a) (7 Points) $F = (b').(a + c').(b + d + e')$

(b) (8 Points) $F = (a + b').(a' + b).(b + c').(b' + c).(a + c').(a' + c)$

3. (a) (10 Points) Write the CNF clauses for the following logic gates: NAND, NOT, NOR.

(b) (10 Points) Write the CNF clauses for the given circuit in Figure 1 and find a satisfying assignment to it (such that z=1).

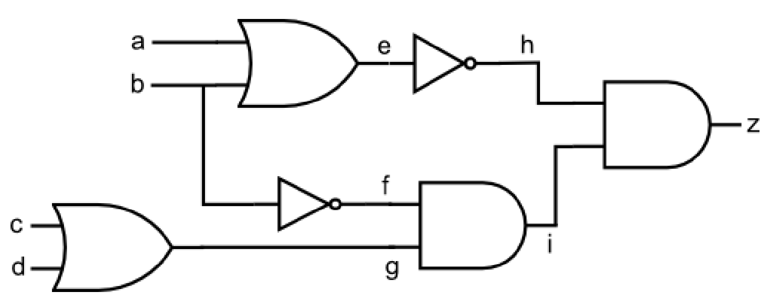

Figure 1: Circuit for SAT check

4. Build a BDD for the following functions:

(a) (5 Points) $F = a + b.c + a'.d.e$ **The order of variables is e, d, c, b, a*

(b) (5 Points) $F = a.b'.c + b.c'.d + c.d'.e + a.d.e'$ **The order of variables is a, b, c, d, e*

(c) (10 Points) $F = a + b + c + a'.b'.c'$ **Build for two different orders: a, b, c and b, c, a*

5. (15 Points) Convert each ordered BDD in Problem 4 to its corresponding ROBDD.

6. (10 Points) Can you re-order the variables so that the generated ROBDD size is smaller than the one in (5-a)?

7. A *Horn clause* is a disjunction of literals with at most one positive literal. A *definite Horn clause* is a Horn clause with **exactly** one positive literal. Horn clauses are used in Horn satisfiability algorithm. The interesting point of Horn satisfiability is that the problem is solvable in polynomial time. We can convert a non-Horn clause to a Horn clause by replacing the complement of some literals in the whole clause. Consider the following CNF formula:

$$(a + b' + c).(a' + c')$$

As it can be seen, the first clause is not a Horn clause. We can re-write this CNF formula to a CNF formula containing Horn clauses by defining $newA = a'$. The converted formula will be:

$$(newA' + b' + c).(newA + c')$$

Now define which of the following CNF formulas are Horn-formulas and which one are definite Horn-formulas. If they are not, re-write them as a definite Horn-formula if possible.

(a) (5 Points) $F = (a).(a' + b + c).(c' + d + e + f)$

(b) (5 Points) $F = (a + b).(a' + b' + c).(a + b' + c' + d).(a + b' + c + d' + e)$

(c) (5 Points) $F = (a + b).(a + b' + d').(a' + b + d' + e)$

(d) (5 Points) $F = (a').(b' + c' + d).(c + e' + f').(b' + e + f' + g')$

8. (30 Points) Write an algorithm to decide if a given set of Horn clauses is satisfiable. What is the complexity of this algorithm? How can you change this algorithm if the input set is a set of definite Horn clauses?