Logic Synthesis & Verification, Fall 2012

National Taiwan University

Problem Set 2

Due on 2012/10/31 before lecture

1 [Cofactor and QBF]

(a) (6%) Given two arbitrary Boolean functions f and g and a Boolean variable v, prove that $(\neg f)_v = \neg(f_v)$ and $(f \ \langle op \rangle \ g)_v = (f_v) \ \langle op \rangle \ (g_v)$ for $\langle op \rangle = \{\vee, \oplus\}$.

(b) (12%) Prove or disprove the following implications:

$$\forall x, \exists y. f(x, y, z) \leftrightarrow \exists y, \forall x. f(x, y, z) \tag{1}$$

$$\forall x. (f(x, y) \wedge g(x, y)) \leftrightarrow (\forall x. f(x, y)) \wedge (\forall x. g(x, y)) \tag{2}$$

$$\forall x. (f(x, y) \vee g(x, y)) \leftrightarrow (\forall x. f(x, y)) \vee (\forall x. g(x, y)) \tag{3}$$

(c) (12%) For an arbitrary Boolean function $f(x_1, \ldots, x_n)$, let a Boolean function $g(x_1, \ldots, x_{n-1})$ satisfy

$$\forall x_n. f(x_1, \ldots, x_n) = f(x_1, \ldots, x_{n-1}, g(x_1, \ldots, x_{n-1})).$$

Express the on-set, off-set, and don't-care-set of g in terms of function f.

2 [BDD Operation]

Let $f = \neg ab \neg c \vee a \neg cd \vee ac \neg d$ and $g = c \oplus d \oplus e$.

(a) (10%) Draw the (shared) ROBDDs of f and g under variable ordering $a < b < c < d < e$ (with a on top).

(b) (10%) Compute the ROBDD of `COMPOSE`(f, c, g).

3 [SAT Solving]

In pseudo Boolean constraint solving, one method is to translate the constraints into a CNF formula for SAT solving. Consider the linear inequality $5x_1 + 3x_2 + x_3 + x_4 + x_5 \geq 6$, where x_i's are Boolean variables and "+" is arithmetic addition.

(a) (10%) Build an ROBDD (variable ordering $x_1 < x_2 < x_3 < x_4 < x_5$) that characterizes the set of feasible solutions to the inequality.

(b) (10%) Treat the above ROBDD as a network of multiplexors and translate it to a CNF formula.

(c) (10%) Show that there exists a linear inequality whose solution-characterizing ROBDD has nodes exponential in the number of variables.

4 [SAT Solving]

Consider SAT solving the CNF formula consisting of the following 8 clauses

$$\begin{aligned}C_1 = (a+b+c), C_2 = (a+b'+d), C_3 = (a+b+c'+d'),\\ C_4 = (b'+c'+d), C_5 = (a+b+d), C_6 = (a'+b'+c),\\ C_7 = (a'+b+d), C_8 = (a+c+d'), C_9 = (a+b'+d').\end{aligned}$$

(a) (10%) Apply implication and conflict-based learning in solving the above CNF formula. Assume the decision order follows a, b, c, and then d; assume each variable is assigned 0 first and then 1. Whenever a conflict occurs, draw the implication graph and enumerate all possible learned clauses under the Unique Implication Point (UIP) principle. (In your implication graphs, annotate each vertex with "`variable = value@decision_level`", e.g., "$b = 0@2$", and annotate each edge with the clause that implication happens.) If there are multiple UIP learned clauses for a conflict, use the one with the UIP closest to the conflict in the implication graph.

(b) (10%) The **resolution** between two clauses $C_i = (C_i^* + x)$ and $C_j = (C_j^* + x')$ (where C_i^* and C_j^* are sub-clauses of C_i and C_j, respectively) is the process of generating their **resolvent** $(C_1^* + C_j^*)$. The resolution is often denoted as

$$\frac{(C_i^* + x) \qquad (C_j^* + x')}{(C_1^* + C_j^*)}$$

A fact is that a learned clause in SAT solving can be derived by a few resolution steps. Show how that the learned clauses of (a) can be obtained by resolution with respect to their implication graphs.