Logic Synthesis & Verification, Fall 2013

National Taiwan University

Problem Set 2

Due on 2013/10/30 before lecture

1 [Cofactor]

(10%) 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 $\langle op \rangle = \{\wedge, \rightarrow\}$.

2 [QBF]

(15%) Prove or disprove the following relations.

(a) $\forall x, \exists y.f(x,y,z) \rightarrow \exists y, \forall x.f(x,y,z)$
(b) $\exists x.(f(x,y) \wedge g(y)) \leftrightarrow (\exists x.f(x,y)) \wedge g(y)$
(c) $\exists x.(f(x,y) \vee g(x,y)) \leftrightarrow (\exists x.f(x,y)) \vee (\exists x.g(x,y))$

3 [Quantifier Elimination]

(20%)

(a) (6%) Let ϕ be a CNF formula over variables $x_1, \ldots, x_n$. Please devise a linear-time procedure to derive a CNF formula $\psi = \forall x_1, \cdots, \forall x_i.\phi$ over variables $x_{i+1}, \ldots, x_n$.
(b) (6%) Repeat (a) but with formulas ϕ and ψ are in DNF and $\psi = \exists x_1, \cdots, \exists x_i.\phi$.
(c) (8%) Assume there is a blackbox procedure that may convert CNF to DNF and vice versa. Please devise an algorithm using this blackbox to perform quantifier elimination for an arbitrary given quantified Boolean formula in the (prenex) CNF form.

4 [ROBDD ITE Operation]

(20%) Let $f = a(b' + d') + a'd'(b' + c')$ and $g = be(c + d) + d'(bc + b'c')$.

(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 $f \wedge g$ using the ITE operation and show your derivation.

5 [ROBDD Minterm Counting]

(15%) Devise a linear-time procedure that counts the number of onset minterms for a given ROBDD.

6 [SAT Solving]

(20%) Consider SAT solving the CNF formula consisting of the following 8 clauses

$$C_1 = (a + b + c), C_2 = (a + b + c' + d'), C_3 = (a + b' + c),$$
$$C_4 = (a + b' + c'), C_5 = (a + c' + d), C_6 = (a' + b + c),$$
$$C_7 = (a' + b' + d), C_8 = (b + d), C_9 = (b' + c + d').$$

(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.