EECS 144 & 244, Fall 2011 **Lee, Roychowdhury, Seshia**

HW 5: Formal Specification & Verification; Symbolic Execution

Assigned: October 19, 2011 *Due: November 2, 2011*

1. (30 points)

   Consider an arbiter circuit, with two input request lines, r_0 and r_1, and two output grant lines z_0 and z_1. The outputs are stored in latches (flip-flops); the latched versions are g_0 and g_1 and form the state of the arbiter along with a single priority bit p which the arbiter uses to alternate its priority assignment between the two input ports in a fair way. Consider the FSM implementing this arbiter with the initial state of the machine, denoted by R_0, and the next-state and output equations as given below:

$$\begin{aligned} R_0 &= p' g_0' g_1' \\ g_0^+ &= r_0(r_1' + p') \\ g_1^+ &= r_1(r_0' + p) \\ p^+ &= r_1' p + r_0 p' \\ z_0 &= g_0 \\ z_1 &= g_1 \end{aligned}$$

   As in the lecture slides, x' denotes the complement of x and x^+ denotes the next-state version of x. (Although we have indicated the outputs z_0 and z_1 above, they do not play a role in any of the questions below.)

   (a) Write down the transition relation δ for this state machine (as a Boolean formula with AND, OR, NOT operators, indicating any simplification that you do) and write the equation expressing R_{k+1} in terms of R_k and δ.

   (b) Perform reachability analysis for this state machine, computing R_1, R_2, ... until termination. Write R_1, R_2, etc. as Boolean formulas free of existential or universal quantifiers.

   You must show all steps in deriving R_1 from R_0 (especially in eliminating the quantifier), but there is no need to show the details for deriving R_2, R_3, etc. Indicate the final R_k. (There is no need to write Boolean formulas as BDDs, you can write any textual Boolean expression including a SOP representation.)

   (c) Is there a reachable state in which g_0 = 1 AND g_1 = 1? Describe an algorithm to check for this automatically.

2. (30 points)

   Express the following properties in linear temporal logic. Explain your answers.

   (a) "Every request must be granted within at most 3 cycles (from the cycle in which the request is raised)"
   Denote "request" by `req`, and the "grant" by `gt`.

   (b) "There are infinitely many transitions from $\neg p$ to p (and vice versa)."

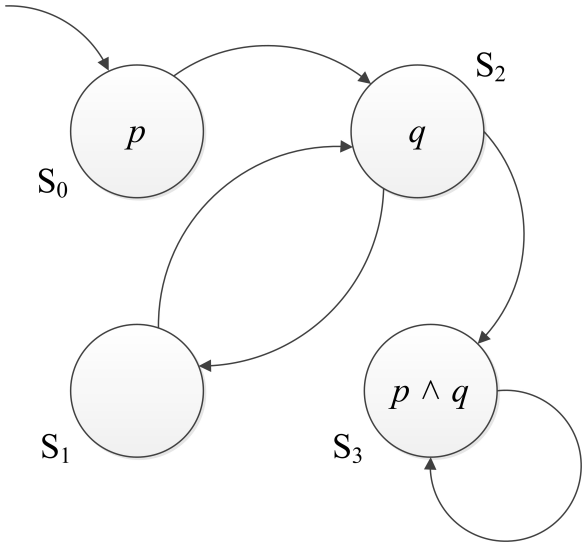
   (c) "The transition from $\neg p$ to p happens at most once."

3. (30 points)

   Consider the Kripke structure M shown in the following figure. S_0 is the initial state. The states are already labeled by propositions that are true in those states. We want to check whether M satisfies the following CTL formula.

$$\mathbf{E}\,\mathbf{F}\,(\mathbf{A}\,\mathbf{G}\,(\mathbf{E}\,\mathbf{G}\,(p \vee q)))$$

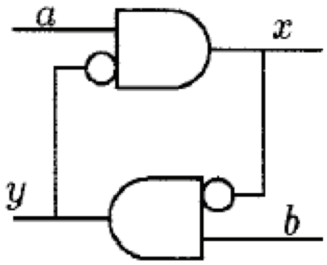
   Given only procedures `CheckEG` and `CheckEU`, describe how you can use them to check the CTL formula. Show the formulas that are satisfied at each state after each sub-formula is checked when model checking is done over the structure of the CTL formula. Does M satisfy the CTL formula?



4. (10 points) Consider the following gate:

Give the characteristic functions for node c in terms of the characteristic functions for nodes a and b.

5. (30 points) The following circuit is studied by [1]:



The primary input is the tuple (a, b). Perform the fixed-point iteration on characteristic functions to determine the characteristic functions of x and y (in terms of a and b). Show all your work.

6. (10 points) Use your result from Question 5 to determine what values of the inputs a and b make the circuit constructive. Show all your work.

References

[1] Thomas R. Shiple, Gerard Berry, and Herve Touati. Constructive analysis of cyclic circuits. In *European conference on Design and Test (DATE)*, pages 328–333. IEEE Computer Society, 1996.