CS 267 – Fall 2009 – Homework Assignment 2

Due Wednesday, November 4th

Do not discuss the problems with anyone other than the instructor.

1. Write the following CTL* formulas using fixpoints: AGAFp, pEU(AGq), EGFp, EFGp.

2. Consider the following transition system $M = (AP, S, R, S_0, L)$ with the set of states $S = \{0, 1, 2, 3\}$, the initial set of states $S_0 = \{0\}$, the transition relation $R = \{(0, 1), (1, 2), (2, 3), (1, 1), (3, 2)\}$, the set of atomic propositions $AP = \{p, q\}$ and the labeling function $L : S \to 2^{AP}$ where $L(0) = \{p\}$, $L(1) = \{q\}$, $L(2) = \{p\}$, and $L(3) = \{p, q\}$.

Based on the above transition system, show the iterative fixpoint computations (show the set of states for each iteration) and the results for the following μ-calculus formulas:

$$\nu z \,.\, (p \,\wedge\, \mathrm{EX}z))$$
$$\mu z \,.\, (q \,\vee\, (p \,\wedge\, \mathrm{AX}z))$$
$$\mu y \,.\, \nu z \,.\, ((q \,\wedge\, \mathrm{EX}z) \,\vee\, \mathrm{EX}y)$$

3. Consider the following transition system $M = (AP, S, R, S_0, L)$ with the set of states $S = \{0, 1\}$, the initial set of states $S_0 = \{0\}$, the transition relation $R = \{(0, 1), (1, 1)\}$ and the set of atomic propositions $AP = \{p\}$ and the labeling function $L : S \to 2^{AP}$ where $L(0) = \emptyset$, $L(1) = \{p\}$. We will use one boolean variable x to encode the states of this transition system as follows: $\{0\} \equiv \neg x$, $\{1\} \equiv x$.

(a) Write the Boolean logic formulas that represent the transition relation R and the set of initial states S_0 of the above transition system for this encoding.

(b) Compute EX(p) using the boolean encoding described above (show the steps of your computation).

(c) Given the variable ordering $x < x'$ draw the BDD for the transition relation R.

4. Given the variable ordering $x_1 < x_2 < x_3$:

(a) Construct the BDDs for the formulas $(x_1 \wedge x_2) \vee x_3$ and $x_1 \wedge \neg x_2$.

(b) Using the BDDs from part **(a)** show the recursive calls for the apply algorithm while computing the disjunction of the above two formulas. Show the resulting BDD.

5. The reader-writer problem is about synchronization of read and write operations on a database. We will assume that the processes are categorized as reader and writer processes. The rules of synchronization are: 1) When a writer process is accessing the database no other process can access the database. 2) When a reader process is accessing the database only other reader processes can access the database. A solution for the reader-writer problem is as follows: (note that the await condition and the following assignment must be executed atomically):

```
nonnegative integer nr;
boolean busy;

initial: nr = 0 and  !busy;

Reader:: while True do
           await (!busy) nr := nr+1;
read:      skip;
           nr := nr-1;
         endwhile

Writer:: while True do
           await (nr = 0 and !busy) busy := true;
write:     skip;
           busy := false;
         endwhile

P = cobegin Reader || Reader || Writer || Writer  coend
```

Write an SMV model for the above specification.

Let R1pc, R2pc, W1pc, W2pc denote the program counters of the four processes. Check the following properties on this system using SMV:

1. $\text{AG}((nr = 0 \vee \neg busy))$
2. $\text{AG}(nr > 0 \Rightarrow (W1pc \neq write \wedge W2pc \neq write))$
3. $\text{AG}((R1pc = read \vee R2pc = read) \Rightarrow nr > 0)$
4. $\text{AG}(R1pc = read \Rightarrow (W1pc \neq write \wedge W2pc \neq write))$
5. $\text{AG}(W1pc = write \Rightarrow (W2pc \neq write \wedge R1pc \neq read \wedge R2pc \neq read))$
6. $\text{AG}(\text{AF}(R1pc = read))$
7. $\text{AG}(\text{AF}(W1pc = write \vee W2pc = write))$
8. $\text{AG}(\text{AF}(nr > 0 \vee busy))$
9. $\text{AG}(W1pc = write \Rightarrow \text{AF}(W1pc \neq write))$
10. $\text{AG}(R1pc = read \Rightarrow \text{AF}(R1pc \neq read))$

Summarize your results by answering the following questions for each property: Does the transition system satisfy the property or is there a counter-example path? How much time did SMV take? How much memory did it use?

How does the verification (or falsification) time change with the increasing number of Reader and Writer processes? (Try it up to at least 5 reader and 5 writer processes).

Turn in your SMV specification and the SMV outputs for the above properties for the 2 reader 2 writer process case.