CS 267 – Spring 2023 – Homework Assignment 2

Due Wednesday May 10th by 5:00pm. Turn in a hard copy either in class or drop in the instructor's mailbox at HFH 2108. Do not discuss the problems with anyone other than the instructor.

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

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

(b) Compute $\mathrm{EX}(p)$ using the boolean encoding described above (show the steps of your computation).

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

2. 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 $\neg x_1 \wedge x_3$.

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

3. Write the following CTL* formulas using fixpoints (i.e., write equivalent formulas in μ-calculus):

(a) $\mathrm{AFAG}p$

(b) $p\mathrm{EU}(\mathrm{AG}q)$

(c) $\mathrm{EGF}p$

(d) $\mathrm{EFG}p$

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

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:

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

5. Consider Dekker's mutual exclusion algorithm given below for two processes:

```
boolean a, b;
integer k;
initial: a = false and b = false and 1 <= k and k <= 2;

while true do
  a := true;
  while b do
    if (k = 2) then
      a := false;
      while (k = 2) do skip; endwhile;
      a := true;
    endif;
  endwhile;
  // critical section
  k := 2;
  a := false;
endwhile
||
while true do
  b := true;
  while a do
    if (k = 1) then
      b := false;
      while (k = 1) do skip; endwhile;
      b := true;
    endif;
  endwhile;
  // critical section
  k := 1;
  b := false;
endwhile
```

(a) Specify Dekker's algorithm in SMV's language.

(b) State the mutual exclusion property for your model in CTL and check if Dekker's algorithms satisfies the mutual exclusion property using SMV. Provide a counter-example (with an explanation) if the property fails.

(c) State the starvation freedom property for your model in CTL and check if Dekker's algorithms satisfies the starvation freedom property. Provide a counter-example (with an explanation) if the property fails.

(d) Add the fairness constraint `FAIRNESS running` and check the starvation freedom property again. Provide a counter-example (with an explanation) if the property fails.

Turn in the printout of your SMV specification and the outputs generated by running SMV on the specification you wrote.