MCSV assignment 1

Kapil Pause

September 25, 2020

Q1

We let atomic proposition X denote that person X is not guilty. Then we have following formulas

$$\begin{aligned} \text{Myshkin} &\iff (\neg\text{Ulyanov} \wedge \text{Lomonosov}) \\ \text{Ulyanov} &\iff (\neg\text{Myshkin} \implies \neg\text{Lomonosov}) \\ \text{Lomonosov} &\iff (\text{Lomonosov} \wedge (\neg\text{Ulyanov} \vee \neg\text{Myshkin}) \end{aligned}$$

checking satisfiability in z3, we find that Ulyanov is true, rest are false. So Ulyanov is not guilty, while Lomonosov and Myshkin are guilty.

Q2

We let $x_0x_1x_2$ be binary representation of x. Since Sir said either transition graph or relation modelling program would suffice, i am only giving transition relation. The program is

```
l_0 :   while True :
l_1 :     if (x % 2==0):
l_2 :       x=x/2
    :     else:
l_3 :       x=(2*x+1)% 8
```

$T(x_0, x_1, x_2, pc, x_0', x_1', x_2', pc')$ is

$$\begin{aligned} &(pc = l_0 \wedge pc' = l_1 \wedge True \wedge same(x_0, x_1, x_2)) \\ \vee &(pc = l_1 \wedge pc' = l_2 \wedge x_2 = 0 \wedge same(x_0, x_1, x_2)) \\ \vee &(pc = l_1 \wedge pc' = l_3 \wedge x_2 = 1 \wedge same(x_0, x_1, x_2)) \\ \vee &(pc = l_2 \wedge pc' = l_0 \wedge x_0' = 0 \wedge x_1' = x_0 \wedge x_2' = x_1) \\ \vee &(pc = l_3 \wedge pc' = l_0 \wedge x_0' = x_1 \wedge x_1' = x_2 \wedge x_2' = 1) \end{aligned}$$

Since in the code, we are not required to encode pc, the transition relation we use is

$$T(x,y) := (x_2 = 0 \wedge y_0 = 0 \wedge y_1 = x_0 \wedge y_2 = x_1) \vee (x_2 = 1 \wedge y_2 = 1 \wedge y_1 = x_2 \wedge y_0 = x_1)$$

After executing Q2a.smt2, we find that all possible states after three iterations are $x = 0, 3$ and 7

After executing Q2b part1.smt2, we find that after 5 iterations, we reach a fixed point. This is verified by checking unsatisfiability of

$$\neg(T5(x,y) \implies T(y,y))$$

Where T5 is defined so that $T5(x,y)$ is True when y is reachable from x after 5 iterations. After executing Q2b part2.smt2, we find that $x = 0, 7$ are only possible fixed points.

Q3

```
LTLSPEC G !(light1=green & light2=green) checks if both lights are never both green.
LTLSPEC G F (light1=green) checks if light1 becomes green infinitely often.
LTLSPEC G !(light1=green & X (light1=red)) checks if light1 becomes yellow before
transitioning to red from green.
SPEC  AG !(light1=green & light2=green) checks if both lights are never both green
SPEC  AG AF (light1=green) checks if light1 becomes green infinitely often.
SPEC  AG !(light1=green & EX (light1=red)) checks if light1 becomes yellow before
transitioning to red from green.
```

Q4

```
LTLSPEC G!(proc0.state=critical&proc1.state=critical) checks if both processes are
never both in critical section
LTLSPEC (G (proc0.state = entering -> F proc0.state = critical)) checks if process
proc0 trying to enter critical section eventually succeeds.
```