HARAMRIT SINGH KHURANA *G01229319*

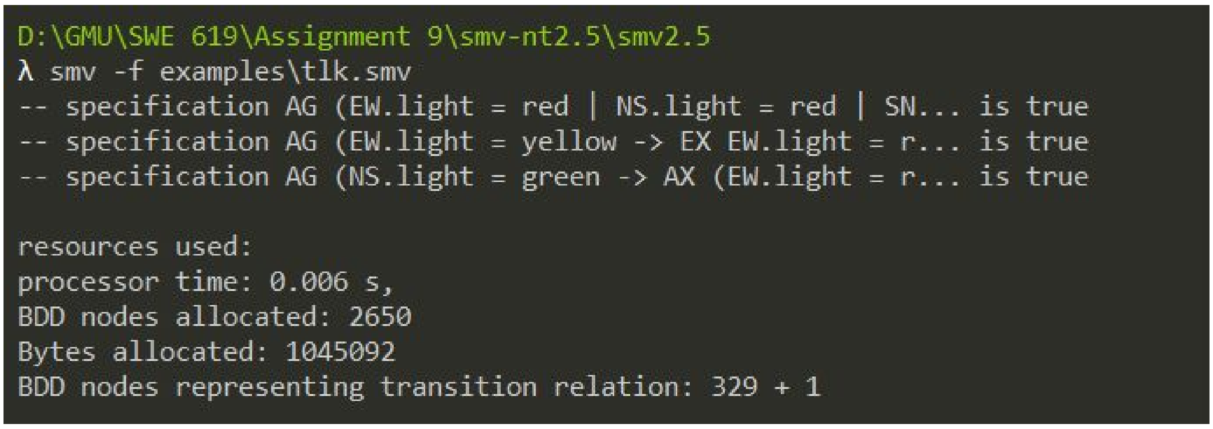
Assignment 9 (Bonus)

SMV File:

```
-- SMV Specification for 4-way intersection
-- Haramrit Singh Khurana

MODULE main
VAR
  EW : direction;
  NS : direction;
  WE : direction;
  SN : direction;

  EWGreen  : process ToGreen(EW, SN, WE, EW);
  NSGreen  : process ToGreen(NS, SN, WE, EW);
  WEGreen  : process ToGreen(WE, SN, NS, EW);
  SNGreen  : process ToGreen(SN, NS, WE, EW);

  EWYellow : process ToYellow(EW);
  NSYellow : process ToYellow(NS);

  EWRed    : process ToRed(EW);
  NSRed    : process ToRed(NS);

SPEC
   AG(EW.light = red | NS.light = red | SN.light = red | WE.light = red)

SPEC
         AG((EW.light = yellow) -> EX (EW.light = red))

SPEC
   AG((NS.light = green ) -> AX (EW.light = red | SN.light = red | WE.light = red))


MODULE direction
VAR
   light : {red, yellow, green};
   leftTurn: boolean;
   rightTurn: boolean;

ASSIGN
   init(light) := red;
   init(leftTurn) := FALSE;
  init(rightTurn) := FALSE;

MODULE ToGreen(dir1, cdir1, cdir2, cdir3)

ASSIGN
  next(dir1.light) :=
    case
     (dir1.light = red) & (cdir1.light = red) & (cdir2.light = red) & (cdir3.light = red) : green;
     TRUE : dir1.light;
    esac;

  next(dir1.leftTurn) :=
         case
          (dir1.light = green) : TRUE;
          TRUE: FALSE;
         esac;

  next(dir1.rightTurn) :=
         case
          (dir1.light = green) : TRUE;
          TRUE: FALSE;
         esac;


MODULE ToYellow(dir1)

ASSIGN
  next(dir1.light) :=
    case
     (dir1.light = green) : yellow;
     TRUE : dir1.light;
    esac;

  next(dir1.leftTurn) :=
         case
          (dir1.light = yellow) : FALSE;
          TRUE: FALSE;
         esac;

  next(dir1.rightTurn) :=
         case
          (dir1.light = yellow) : TRUE;
          TRUE: FALSE;
         esac;
```

```
MODULE ToRed(dir1)

ASSIGN
  next(dir1.light) :=
    case
     (dir1.light = yellow) : red;
     TRUE : dir1.light;
    esac;
  next(dir1.leftTurn) :=
          case
           (dir1.light = red) : FALSE;
           TRUE: FALSE;
          esac;

  next(dir1.rightTurn) :=
          case
           (dir1.light = red) : TRUE;
           TRUE: FALSE;
          esac;
```

Output:



```
D:\GMU\SWE 619\Assignment 9\smv-nt2.5\smv2.5
λ smv -f examples\tlk.smv
-- specification AG (EW.light = red | NS.light = red | SN... is true
-- specification AG (EW.light = yellow -> EX EW.light = r... is true
-- specification AG (NS.light = green -> AX (EW.light = r... is true

resources used:
processor time: 0.006 s,
BDD nodes allocated: 2650
Bytes allocated: 1045092
BDD nodes representing transition relation: 329 + 1
```

Assumptions:

So, in the above model, I have considered a 4-way intersection which implies 4 directions - NS, SN, WE, EW.
For each direction, I am assuming 3 lanes - one for going straight (middle lane), one for going right (right lane) and one for going left (left lane).
Based on my observation, here in the US, at any given point of time, generally right turns in any given direction are YIELD signs, i.e. vehicles can take that turn looking at other cars and pedestrians. Hence, I have allowed rightTurn for each direction to be ON for most of the time.
The problematic ones are the middle lane and the left lane. These lanes need to strictly stop at the red signal and hence leftTurn becomes false as soon as the signal turns red.