Formal Methods Project: "*Crossroads*"

Nika Pona, EMCL 2014/2015, ID: 13025

I chose to model the traffic lights controller specified in the attached file *crossroads.pdf*. My program is contained in the file *project.smv*. It consists of two modules: the obligatory **main** module and the **controller** module called from **main** to create states to model West-East (*we*) and North-East (*ns*) controllers respectively. Intuitively, each controller is a function from other road's state and their traffic lights and models the possible situations on each road.

```
MODULE main
VAR
we :  controller(ns.state,ns.green);
ns :  controller(we.state,we.green);
ASSIGN
init(ns.green) := TRUE;
init(we.green) := FALSE;
```

In the module **controller** I define the possible states of the road and traffic lights following the specifications.

```
MODULE controller(other_state,other_green)
VAR
state : {detected,waiting,passing};
green : boolean;
ASSIGN
next(state) :=
case
  state in {detected,waiting} & green : passing;
  state in {detected,waiting} & !green: waiting;
  state = passing : detected;
esac;
next(green) :=
case
  state = detected & other_state in {detected,waiting} & other_green : FALS
  state = detected : TRUE;
  state = passing & other_state = passing & other_green : TRUE;
  state = passing : FALSE;
  state = waiting & other_state in {detected,waiting} & other_green: FALSE;
  state = waiting : TRUE;
esac;
```

I test the following properties:

- If there is a car waiting on the West-East road, it will eventually pass:

```
AG (we.state = waiting -> AF we.state = passing)
```

- If there is a car waiting on the North-South road, it will eventually pass:

```
AG (ns.state = waiting -> AF ns.state = passing)
```

- It is never the case that both traffic lights are green:

```
AG !(ns.green = TRUE & we.green = TRUE)
```