Modelling and Analysis of a Cyber-Physical System

Cyber-Physical Programming — Practical Assignment 1

Melânia Pereira
pg47520@alunos.uminho.pt

Paulo R. Pereira
pg47554@alunos.uminho.pt

May 16, 2022

Abstract

The first goal of the assignment is to model and analyse a system that ensures the correct functioning of traffic lights at a T-junction (where the it connects a "major" and a "minor" road). This system of traffic light works reasonably well under the assumption that one of the roads has more traffic than the other. But such an assumption is often too strong: it may be the case that both roads have the same amount of traffic, or even that their traffic flow varies drastically throughout the day. Therefore, the second part of this assignment aims to address precisely this problem which is well-known to have significant impact in the economy and the environment[1]. To this effect, we assume that each traffic light has a smart sensor attached to it. The sensor informs whether the traffic near the light is high, low, or simply non-existent.

First Part

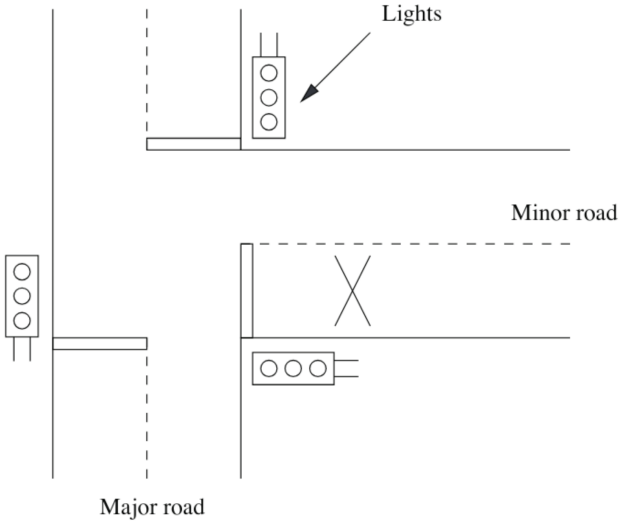

Figure 1: T-junction scenario.

In this scenario vehicles drive on the left side of the road and the cross in the picture represents a sensor that tells whether a car is waiting in the minor road or not. In order to guarantee a reasonable traffic flow, the system has the following constraints:

[1] https://ourworld.unu.edu/en/green-idea-self-organizing-traffic-signals

- The lights on the major road will be always set on green, and red on the minor road unless a vehicle is detected by the sensor.
- In the latter case, the lights will switch in the standard manner and allow traffic to leave the minor road. After a suitable time interval (30s), the lights will revert to their default position so that traffic can flow on the major road again.
- Finally, as soon as a vehicle is detected by the sensor the latter is disabled until the minor-road lights are on red again.

The system also respects the following temporal constraints:

1. Interim lights stay on for 5s.
2. There exists 1s delay between switching one light off and the other on.
3. The major-road light must stay on green for at least 30s in each polling cycle, but must respond to the sensor immediately after that.

1 Modelling in UPPAAL

First, we need to model the sensor.

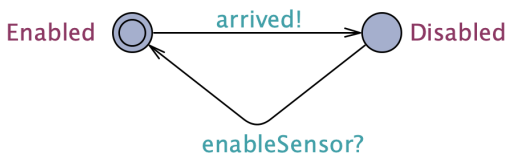

Figure 2: Sensor Model

Sensor only have 2 states and it starts enabled. There are no clocks associated and the set of actions is when a car arrives and it goes off and when the minor-road red light goes on and it reactivates.

N.B.: *enableSensor* was declared as an **urgent** channel. This means that no delay is allowed in this synchronisation. As we'll see later, in the minor-road traffic lights, when the red light is on, the sensor must be reactivated (according to the third point) and it occurs immediately (because it was declared as urgent).

Now, modelling the traffic lights is quite more complex. Let's talk about the the major traffic lights first, by seeing the UPPAAL model in the following figure.

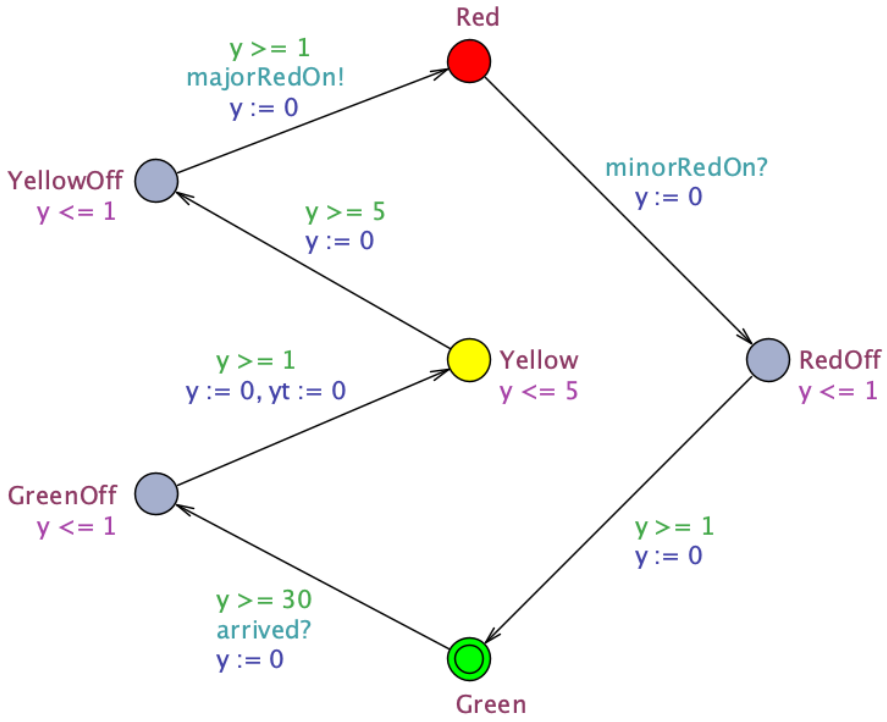

Figure 3: Major Traffic Lights Model

It's obvious we'll need at least three states — Red, Yellow and Green. However, considering the second temporal constraint *There exists 1s delay between switching one light off and the other on*, three more states were added — RedOff, YellowOff and GreenOff. These names are very suggestive, e.g., the transition between Green and Yellow means switching green light off and then turn yellow on.

We also need a clock (or at least one clock, we'll get there). That's quite obvious because we have some temporal constraints which we'll analyze soon.

The set of actions is also very intuitive. We have already seen that the sensor transmits information through channel *arrived*. So, major traffic light need to receive it! It will only happen when the green light is on and after (at least) 30s of green lighting, respecting the third temporal constraint. Also, there must be communication between the traffic light — one only goes green after the other goes red. So, each traffic light must send to the other the information that its red light is on! Only then the other traffic light can go green.

Now, according to the second temporal constraint, we need to guarantee the 1 second delay between lights' switching, i.e., in every intermediate state (*RedOff*, *YellowOff* and *GreenOff*) clock y cannot exceed 1 second.[2] According to the first temporal constraint, clock y cannot exceed 5 seconds in the interim light (or yellow) state. For the remaining states, the invariant is the constant function $\underline{true}$.

Some important observations:

- The transition $Green \xrightarrow{z \geq 30,\ arrived?,\ \{y\}} GreenOff$ can only happen if there is a car waiting in the minor road (i.e., if the sensor sends that information through channel *arrived*) and if $z \geq 30$, respecting the third temporal constraint.

[2]The expression "clock cannot exceed" can be used because at every transition, the clock is reset.

- When the transition $YellowOff \xrightarrow{y \geq 1,\ majorRedOn!,\ \{y\}} Red$ occurs, it sends information through channel *majorRedOn* saying "Hey, I'm going red!" This will synchronize with the minor traffic lights — minor traffic lights receives that information when the red light is on, so it says "Ok, now I can go green!"
- The transition $Red \xrightarrow{minorRedOn?,\ \{y\}} RedOff$ needs this synchronization to ensures that it doesn't go green when the minor traffic light is still green.
- There is another clock (yt). This clock is only to prove the property of how long yellow light is on.

Finally, the mechanism for the minor traffic lights is analogous — just swap the two channels *minorRedOn* and *majorRedOn*, remove the synchronization in the transition $Green \xrightarrow{z \geq 30,\ \{z\}} RedOff$ and change the clocks to z and zt. But there's only one exception. Let's look to the model and find the exception.

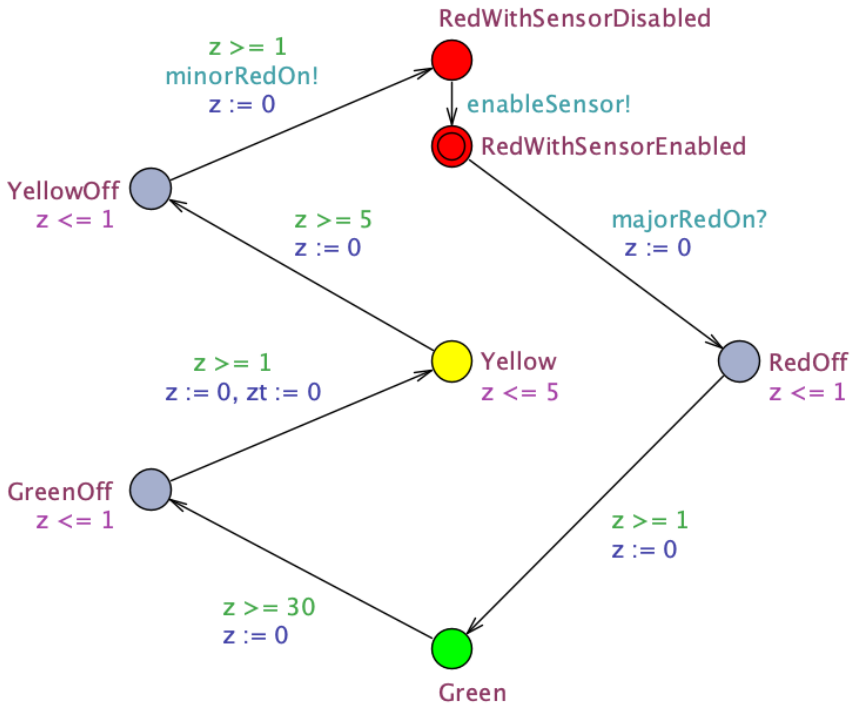

Figure 4: Minor Traffic Lights Model

The exception is the new state *RedWithSensorEnabled*. This one is to respect the third point — as soon as a vehicle is detected by the sensor the latter is disabled until the minor-road lights are on red again. So, the synchronization via channel *enableSensor* happens without delay (as explained before) because this channel was declared urgent. Thus, once minor-road light goes red, the sensor is reactivated.

2 System properties expressed in CTL and tested in UPPAAL

2.1 Reachability properties

Property 1 *The minor-road light can go green.*

$$E\Diamond \ MinorTrafficLights.Green$$

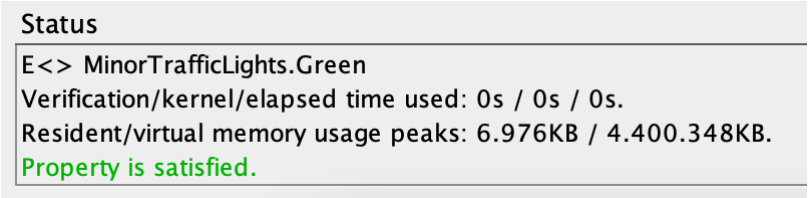

Figure 5: Property 1 is valid.

Property 2 *The major-road light can go red.*

$$E\Diamond \ MajorTrafficLights.Red$$

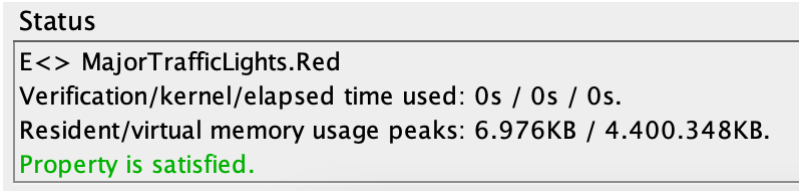

Figure 6: Property 2 is valid.

2.2 Safety properties

Property 3 *The system never enters in a deadlock state.*

$$A\Box \ \neg \ deadlock$$

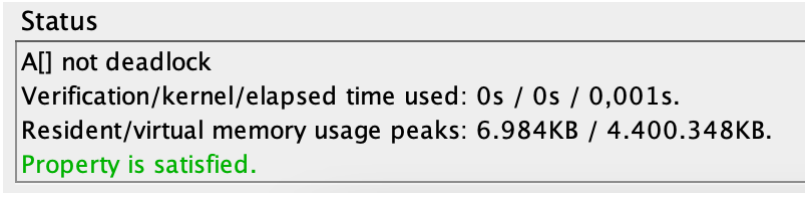

Figure 7: Property 3 is valid.

Property 4 *The minor-road and major-road lights cannot be green at the same time.*

$$A\Box \ \neg \ (MajorTrafficLights.Green \wedge MinorTrafficLights.Green)$$

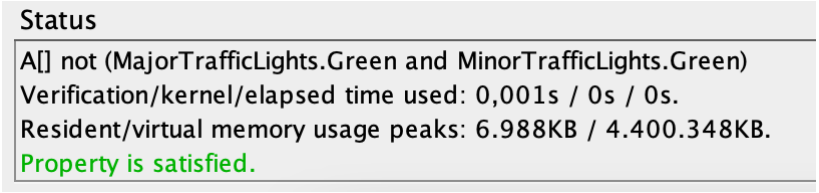

Figure 8: Property 4 is valid.

2.3 Liveness properties

Property 5 *If there are cars waiting they will eventually have green light.*
This property is not so obvious. One could easily say *Sensor.Disabled* $\rightsquigarrow$ *MinorTrafficLights.Green*, i.e., saying that there are cars waiting is the same as saying that the sensor is disabled. However, let's see the following trace:

1. Major light starts green;
2. Sensor recognizes cars;
3. Major green light off;
4. Sensor disabled;
5. Meanwhile, major goes red and minor goes green;
6. Cars pass and minor gets red and sensor is reactivated;
7. No more cars in the minor road — minor green light will no more go on.

So, the fact that the sensor is disabled does not mean that there are cars waiting. We know that there are cars waiting if the major green light goes off.

$$\textit{MajorTrafficLights.GreenOff} \rightsquigarrow \textit{MinorTrafficLights.Green}$$

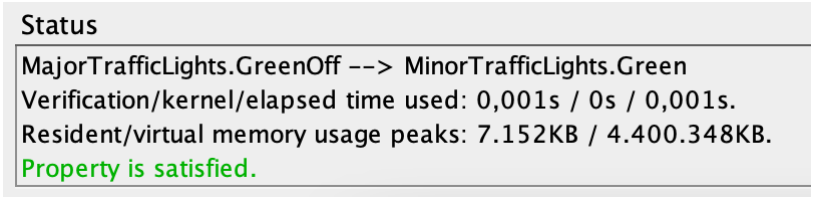

Figure 9: Property 5 is valid.

N.B.: The previous properties were requested in the statement. The following new properties were added.

Property 6 *If major-road yellow light is on, then the minor-road red light is on (and vice-versa).* (Safety)

$$A\Box \ \textit{MajorTrafficLights.Yellow} \Rightarrow \textit{MinorTrafficLights.Red or MinorTrafficLights.RedWithSensorEnabled}$$

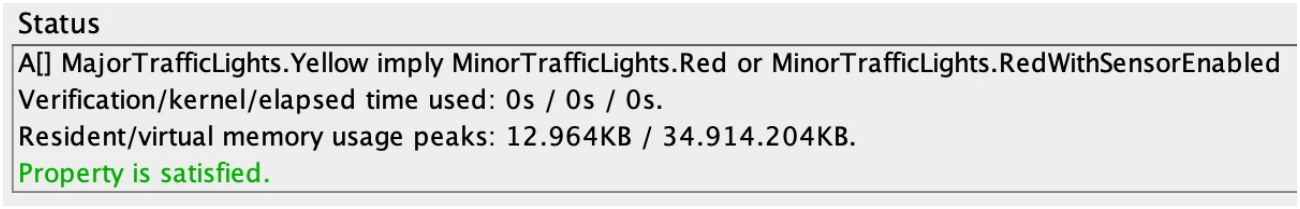

Figure 10: Property 6 is valid.

Property 7 *If yellow light is on, then the red light will be on in exactly 6 seconds.* (Reachability)

$$\textit{MajorTrafficLights.Yellow} \rightsquigarrow \textit{(MajorTrafficLights.Red imply yt==6)}$$

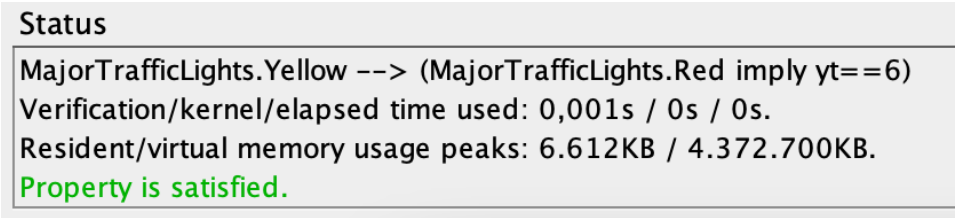

Figure 11: Property 7 is valid for major-road traffic lights.

$$\textit{MinorTrafficLights.Yellow} \rightsquigarrow \textit{(MinorTrafficLights.RedWithSensorDisabled imply zt==6)}$$

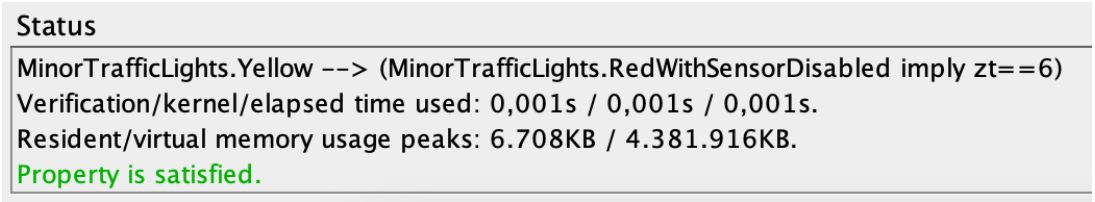

Figure 12: Property 7 is valid for minor-road traffic lights.

SECOND PART

The previous system of traffic lights works reasonably well under the assumption that one of the roads has more traffic than the other. But such an assumption is often too strong: it may be the case that both roads have the same amount of traffic, or even that their traffic flow varies drastically throughout the day. So, the goal of this second part is to achieve that reality. For that, we assume that each traffic light has a smart sensor attached to it. The sensor informs whether the traffic near the light is high, low, or simply non-existent.

1 Adapting the Model in UPPAAL

As said above, we'll need to model the smart sensors of each road.

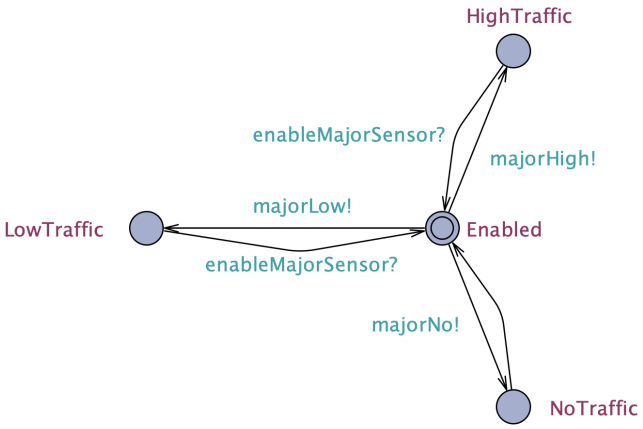

Figure 13: Major Smart Sensor Model

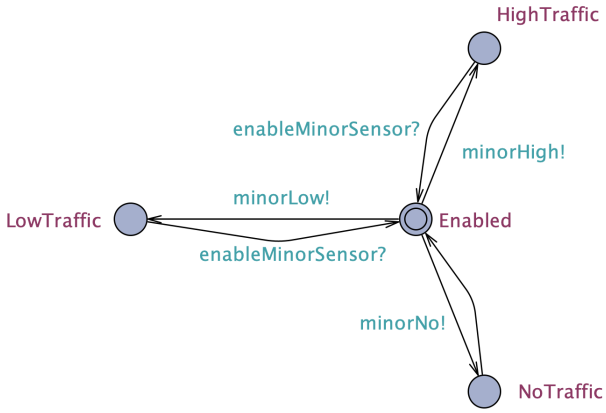

Figure 14: Minor Smart Sensor Model

So, both sensors have 3 main states — one for low traffic, another for high traffic and another one for the initial (enabled) state. There is a 4^{th} state, *NoTraffic*, which is abstract the same of the *Enabled* state but can be useful for testing some properties. The actions for both models refers to the synchronizations of the traffic lights models when they have to switch lights, as we'll see soon. There are no clocks associated and the invariant for all states is the constant function *true*.

Now, let's talk about the model of the traffic lights. It will be similar to the previous models with the exceptions that both minor and major model will have restrictions in the transition between *Green* and *GreenOff* — this is necessary to ensure the expected reality mentioned above.

N.B.: not all the details will be explained because most of the details are the same as the previous model, except the details that will be explained below.

So, let's take a look to the new major road traffic lights model.

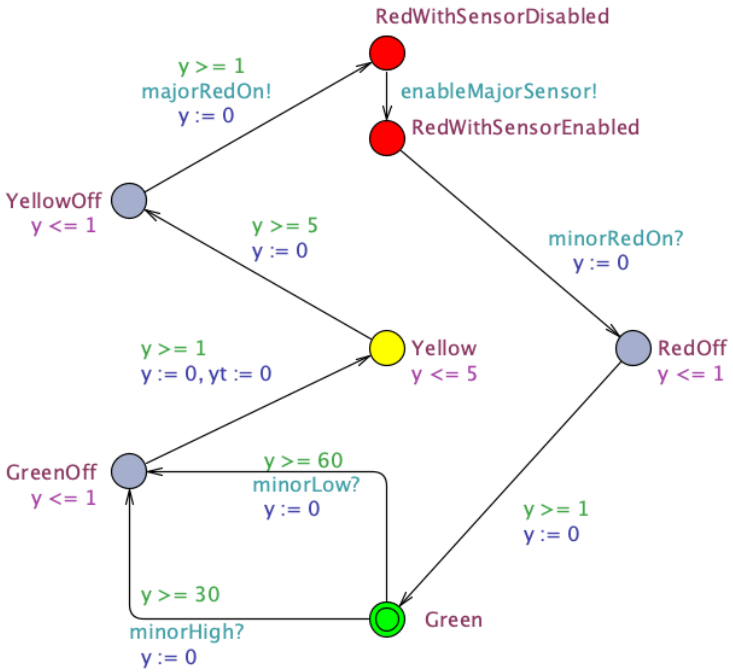

Figure 15: Major Traffic Lights New Model

As we see, it starts on *Green*, and it can only change if it receives information through channels *minorLow* and *minorHigh* — this means that it can only go to *GreenOff* state if some car is waiting in the minor road — and, depending on the traffic intensity it will be green for at least 30 or 60 seconds. The minor road model is similar to this one.

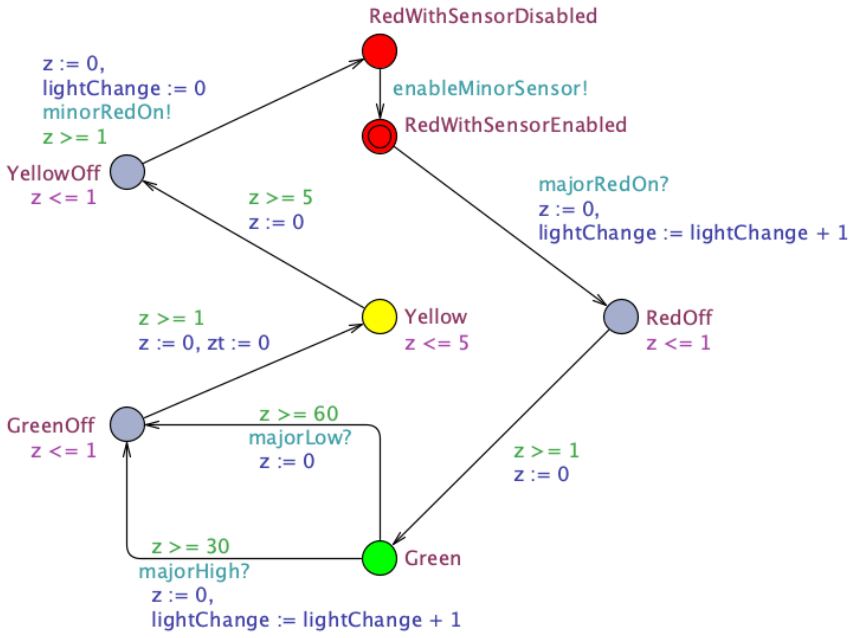

Figure 16: Minor Traffic Lights New Model

There are, of course, some changes — swapping channels to the correspondent ones, changing the initial state and the addition of the variable *lightChange*. This variable is used to prove properties related to

the number of light changes after the light is red.

One important thing to highlight is that both roads' traffic lights model have restrictions to switch off the green light. This makes possible the real purpose of the system — the fact that traffic lights "play" with each other to maintain a smooth and functional traffic.

Now, to understand better how this model works, let's see some new properties.

2 Same and new properties expressed in CTL and tested in UPPAAL

The same properties of the (previous) first model continue valid (which is expected).

Now, according to the new model, it is important to see some properties that say something about the efficiency of the new system.

Property 9 *If the rightmost sensor always detects high traffic and the others detect no traffic at all, then we will observe at most one change in the traffic light*

$$A\Box\ (SensorMinorRoad.HighTraffic \wedge SensorMajorRoad.NoTraffic) \Rightarrow 1 \geq lightChange$$

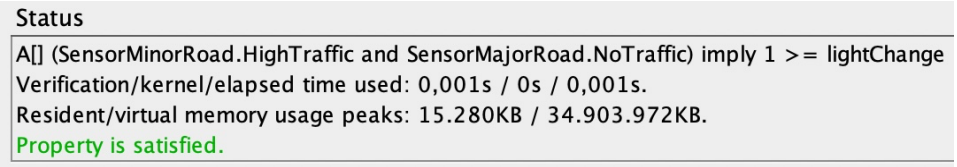

Figure 17: Property 9 is valid.

Property 10 *If no car uses the major-road, the minor-road is always green.* (Liveness)

$$A\Box\ SensorMajorRoad.NoTraffic \Rightarrow MinorTrafficLights.Green$$

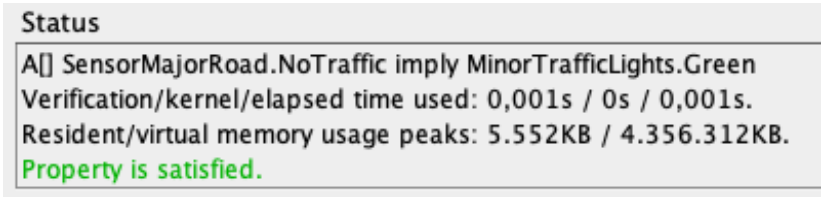

Figure 18: Property 10 is valid.

Property 11 *If the minor-road sensor outputs something and the other sensor outputs nothing then the minor-road traffic light is on green at least until the sensors provide new information.* (Liveness)

$$A\Diamond\ SensorMinorRoad.LowTraffic \vee SensorMinorRoad.HighTraffic \Rightarrow MinorTrafficLights.Green$$

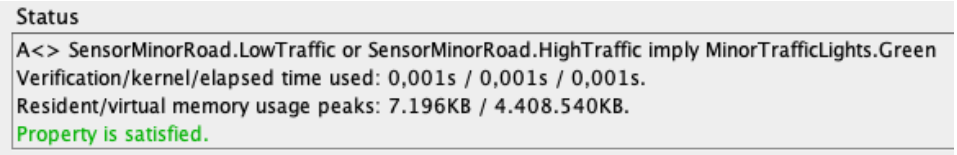

Figure 19: Property 11 is valid.

Conclusion

The new model is far better than the first one. We can conclude that from the new properties. For example, if the minor-road sensor outputs high traffic and the other sensors outputs no traffic then the minor-road traffic light is always on green until the sensors provide new information (which can be checked by properties 10 and 11). This scenario in the first model was very inefficient because the major traffic light would be repeatedly green for 30 seconds after 30 seconds of minor green light, even if no car was is the major road.

Thus, we are proud of the new model and we are sure it achieves the reality of road traffic.