Software Architecture and Calculi - Assignment 1

Design and analysis of a cyber-physical system

A85700 Pedro Costa

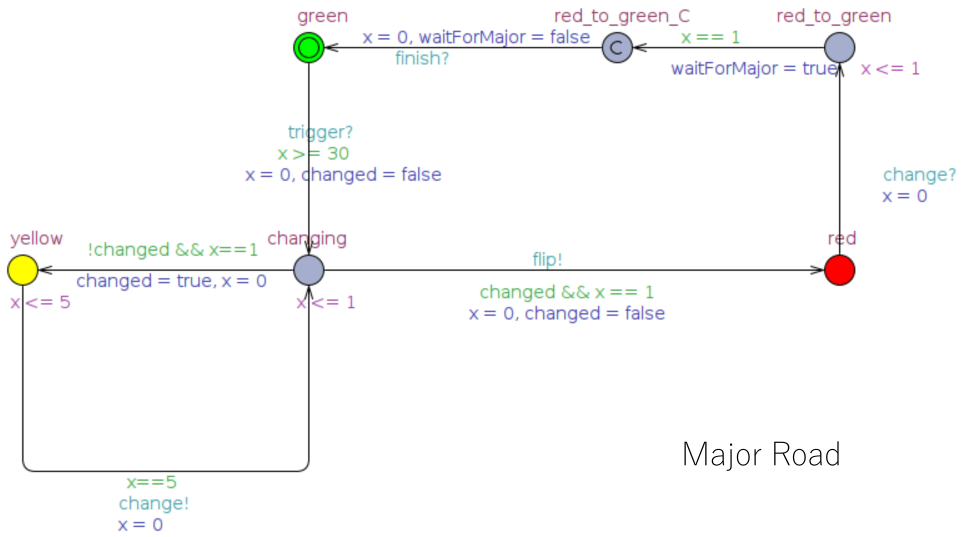
A85954 Luís Ribeiro

First Part

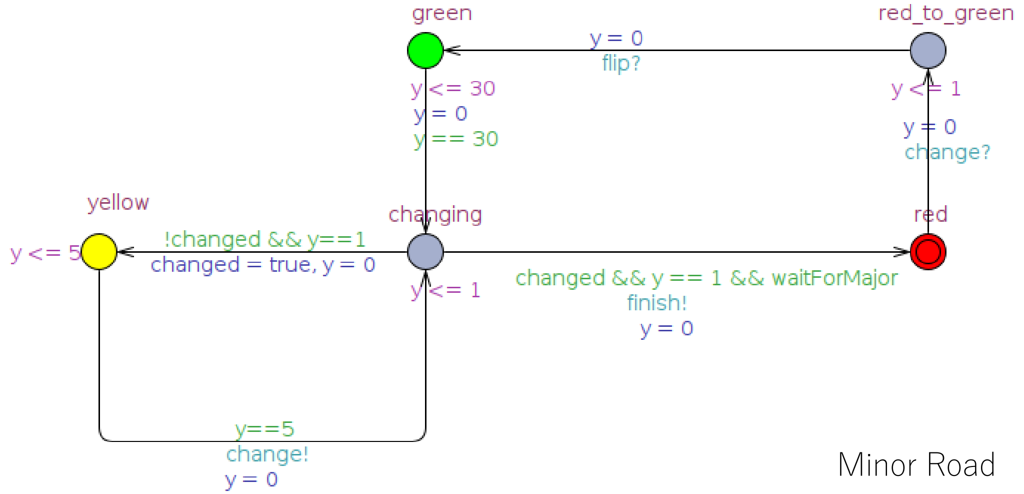
- **Design Choices**
  - Approach where the concept of traffic light was abstracted by its road.
  - The sensor dictates how the system evolves.
  - Our second approach – The traffic light approach. What led us to choose the other approach?
- **UPPAAL Models**
- **Properties**
  - Background, Reachability, Safety and Liveness.
  - Extras.

First Part

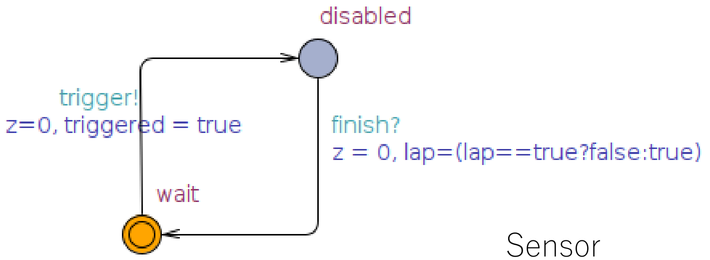
UPPAAL Models



Major Road



Minor Road



Sensor

First Part

Properties

Reachability

The minor-road light can go green:

✓ `E<> MinorRoad.green`

The major-road ligth can go red:

✓ `E<> MajorRoad.red`

Safety

The system never enters in a deadlock state:

✓ `A[] !deadlock`

The minor-road and major-road green lights can not be on at the same time:

✓ `A[] !(MajorRoad.green and MinorRoad.green)`

Liveness

If there are cars waiting they will eventually have green light:

✓ `MajorRoad.red --> MajorRoad.green`

✓ `Sensor.disabled and MinorRoad.red --> MinorRoad.green`

Extra

The sensor can only be in a reading state while the traffic light on minor-road is red:

✓ `A[] !(MinorRoad.green and Sensor.wait)`

The sensor can only be in a post-reading state while the traffic light on major-road is red:

✓ `A[] !(MajorRoad.green and Sensor.disabled)`

Disabled sensor only comes back to disabled after being in a wait state:

✓ `Sensor.disabled and lap --> (!lap)?(Sensor.disabled):(true)`

✓ `Sensor.disabled and !lap --> (lap)?(Sensor.disabled):(true)`

The amount of time that passes between green lights on major-road is 44 seconds:

✓ `MajorRoad.changing --> MajorRoad.red_to_green_C and Sensor.z == 44`

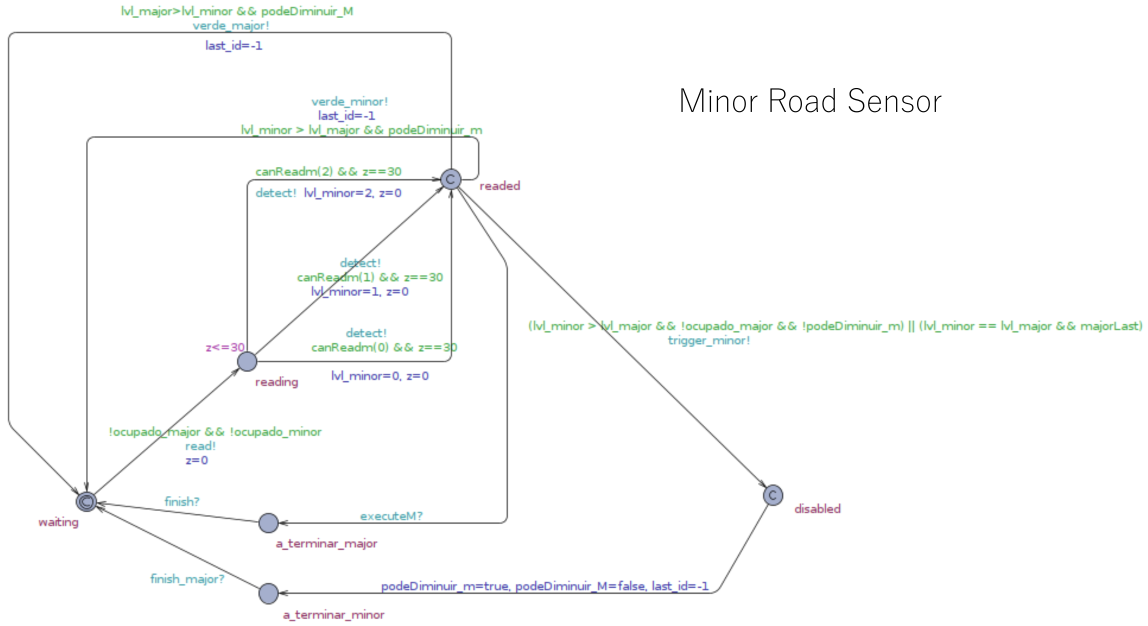
Second Part

- **Design Choices**
  - Different approach and why – Both Major Road traffic lights can't be represented as one.
  - Declare each traffic level by a corresponding integer – 0->none, 1->low, 2->high.
  - First approach (non-synchronized) – Problems and doubts.
  - Second and final approach (synchronized) – Simplifying the sensor's templates and dealing with fairness.
- **UPPAAL Models**
  - First approach.
  - Second approach – Before and after SynchronizedS's template.
- **Benchmarking Properties**
  - Reachability, Safety and Liveness.
  - Valorization properties.

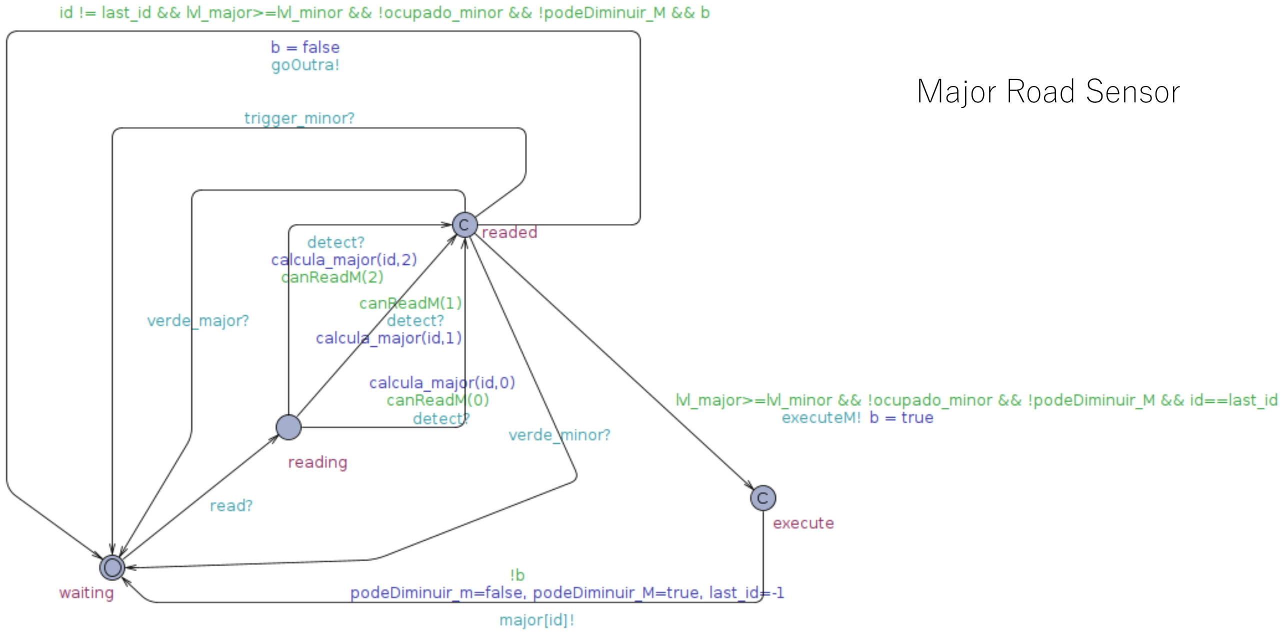
Second Part

UPPAAL Models – Before Synchronize Sensor Template

Minor Road Sensor



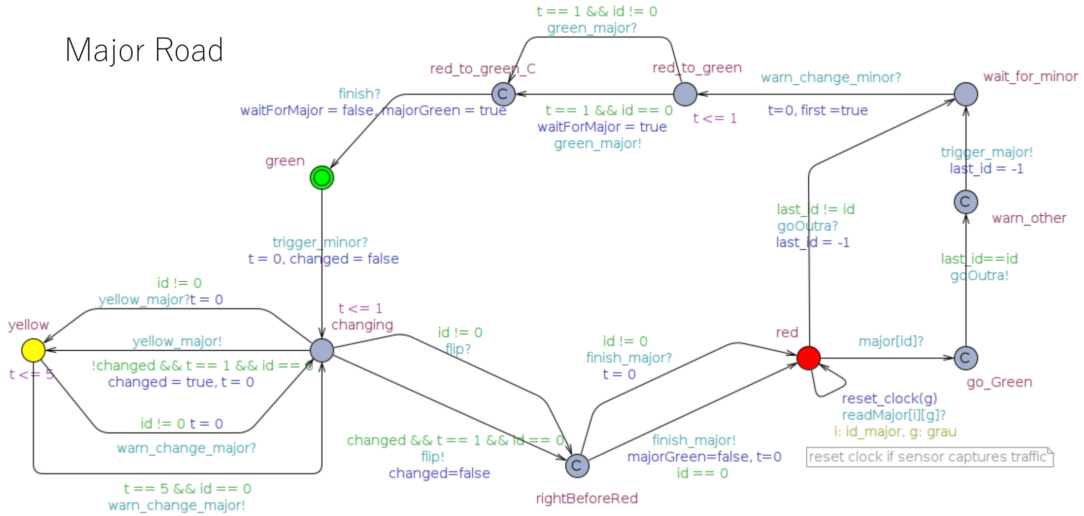
Major Road Sensor



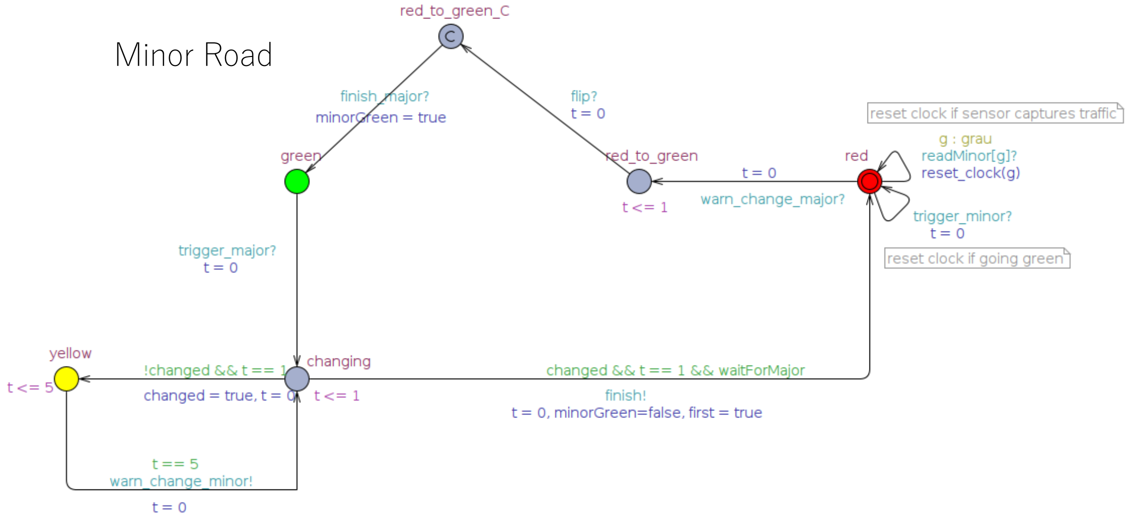
Second Part

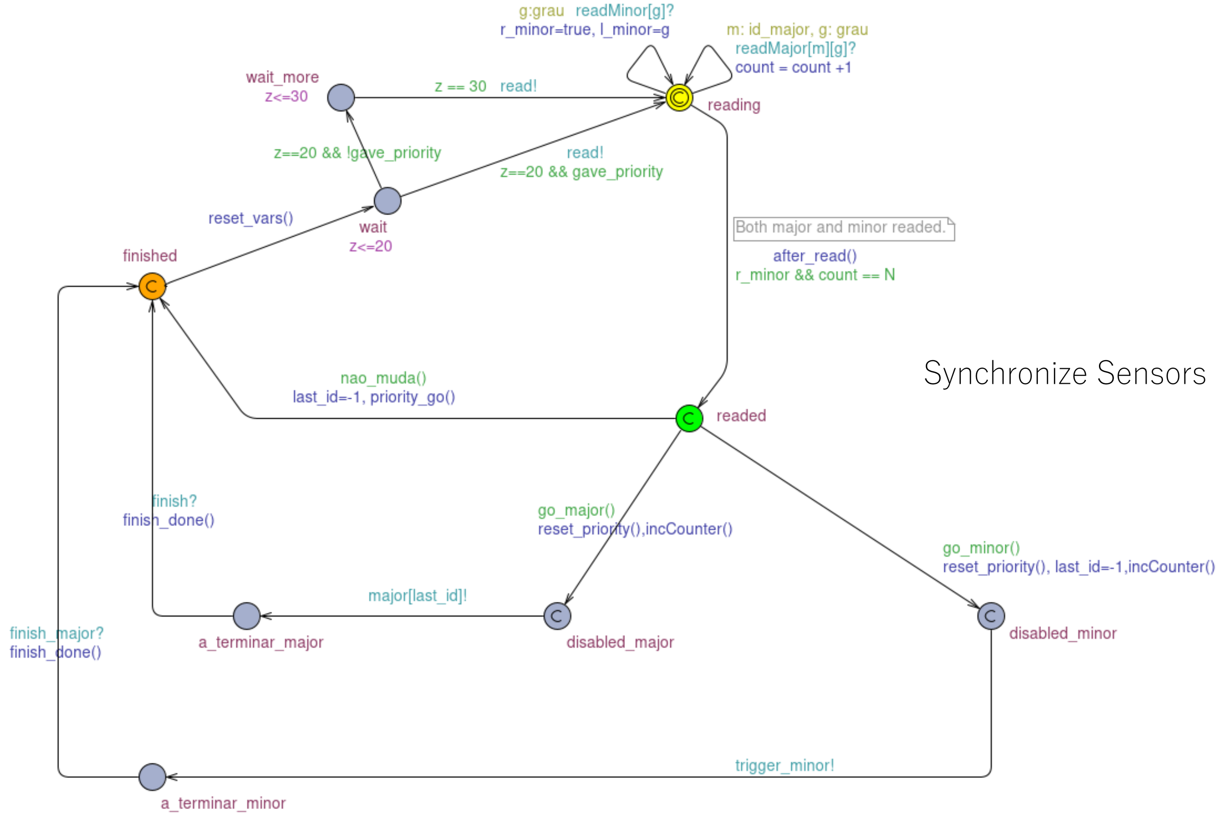
UPPAAL Models – Last Approach

Major Road

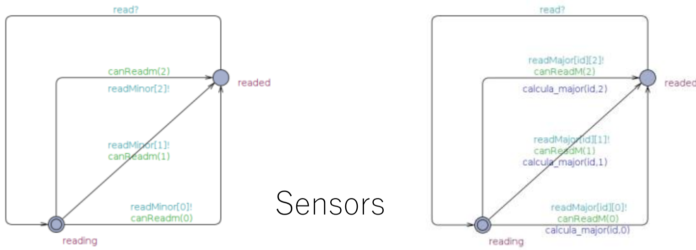


Minor Road





Synchronize Sensors



Sensors

Second Part

New Properties defined

Safety

Both major-road traffic lights are synchronized (both at green at the same time):

- ✓ `A[] (MajorLight(0).green imply MajorLight(1).green)`
- ✓ `A[] (MajorLight(1).green imply MajorLight(0).green)`

Liveness

MajorRoad will be green if it has at least low traffic, even though traffic in MinorRoad is higher:

- ✓ `MajorLight(0).red && MajorLight(1).red && lvl_major < lvl_minor && lvl_major > 0 --> MajorLight(0).green && MajorLight(1).green`

MinorRoad will be green if it has at least low traffic, even though traffic in MajorRoad is higher:

- ✓ `MinorLight.red && lvl_minor < lvl_major && lvl_minor > 0 --> MinorLight.green`

Benchmarking

If the MinorRoad sensor is always detecting high traffic and the other sensors do not detect any traffic, then we observe a maximum of 1 signal exchange:

- ✓ `lvl_minor == 2 && lvl_major == 0 && flipCounter <= 1 --> lvl_minor == 2 && lvl_major == 0 && flipCounter <= 1`

If both roads have the same traffic level then the lights will alternate constantly:

- ✓ `lvl_minor == lvl_major && SynchronizeS.readed && lvl_minor > 0 && majorGreen --> lvl_minor == lvl_major && lvl_minor > 0 && !majorGreen`

If a road has traffic, the waiting time before turning green is less than 60 seconds:

- ✓ `A[] forall(i:id_major) lvl_major>0 && MajorLight(i).red imply MajorLight(i).t<=60`
- ✓ `A[] lvl_minor > 0 && MinorLight.red imply MinorLight.t <= 60`

The time passed between two traffic reads is at most 37 (30 on green plus 7 on lights changing) seconds:

- ✓ `A[] transiction_time <= 37`

If one road gets permission to release traffic (the other road gave permission to allow traffic flow), the transiction time is always 27 (20 on green plus 7 on lights changing) seconds, before reading again:

- ✓ `A[] (SynchronizeS.gave_priority && SynchronizeS.reading) imply transiction_time == 27`

Conclusion

- **Modelling Advantages**
  - Why modelling time-critical systems?
  - The need of providing a smooth experience to the drivers by exploring fairness.
- **Difficulties**
  - Debate before implementing.
  - Making decisions.
  - Concerns about modelling more complex problems.
- **Final thoughts.**