



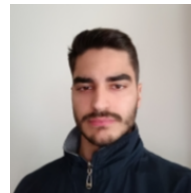
Universidade do Minho
Escola de Engenharia

Programação Ciber-Física

2023/2024

Trabalho Prático 1

Ricardo Lopes Santos Silva :: pg54188; Afonso Xavier Cardoso Marques :: pg53601



I. INTRODUÇÃO

Este trabalho prático centra-se na modelagem e análise de um sistema ciber-físico relacionado com o funcionamento dos semáforos numa interseção em forma de T. O objetivo principal é desenvolver um modelo abrangente que represente de forma precisa o comportamento e as interações dos componentes do sistema, incluindo as estradas principais e secundárias, bem como os semáforos correspondentes. Através da utilização de técnicas de modelagem formal e metodologias de análise, a tarefa visa investigar a funcionalidade e eficácia do sistema ciber-físico na garantia de um fluxo de tráfego seguro e eficiente na interseção.

A seguinte imagem representa o cenário de estudo que consiste numa interseção em forma de T com uma estrada principal (Major Road) e dois semáforos e uma estrada secundária (Minor Road) com um sensor e um semáforo. Neste cenário, os carros conduzem do lado esquerdo.

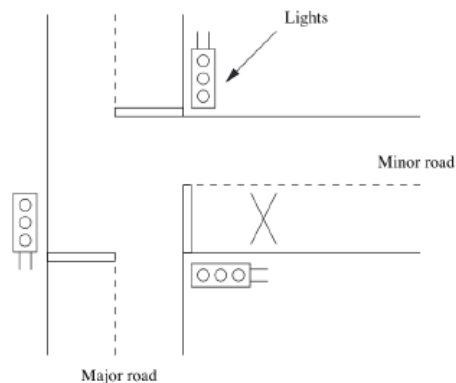


Fig. 1: Cenário em estudo

II. PARTE 1

A. Contextualização

O primeiro objetivo deste trabalho é modelar e analisar o sistema em causa de forma a garantir o funcionamento correto dos semáforos na interseção.

Para garantir um fluxo de tráfego razoável, o sistema tem as seguintes restrições:

- 1) Os semáforos na Major Road estão sempre verdes, e sempre vermelhos na Minor Road, a menos que um veículo seja detetado pelo sensor.
- 2) Caso o sensor seja ativado, os semáforos mudarão de acordo com a ordem normal e permitirão que o tráfego saia da Minor Road. Após um intervalo de tempo adequado (30 segundos), os semáforos voltarão à posição padrão para que o tráfego possa fluir na Major Road novamente.
- 3) Finalmente, assim que um veículo for detetado pelo sensor, este é desativado até que os semáforos da Minor Road voltem a estar vermelhos.

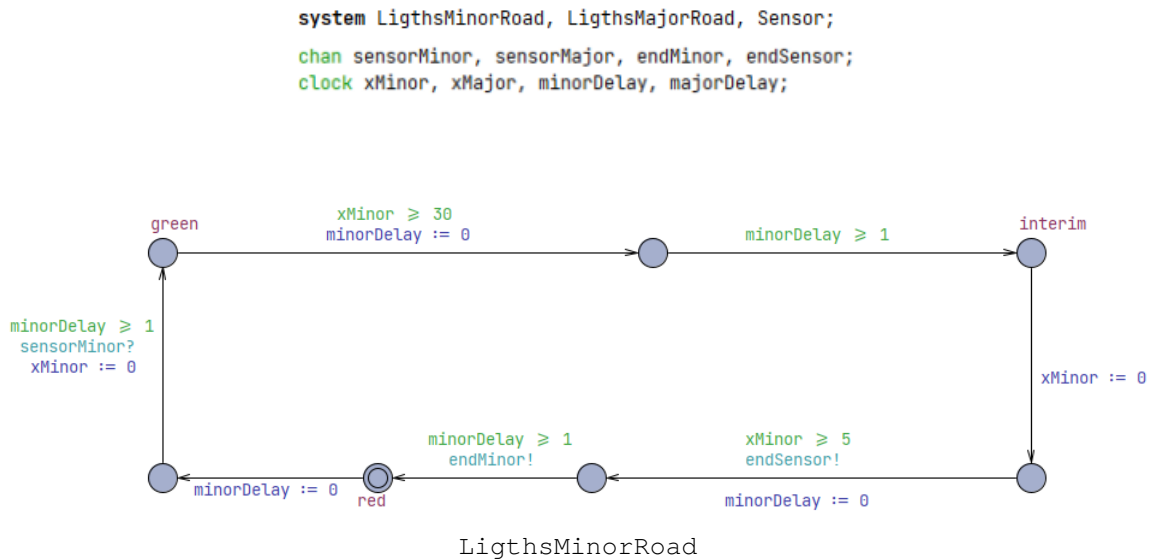
O sistema também respeita as seguintes restrições temporais:

- 1) Os semáforos intermediários (com luz amarela) permanecem ligados por 5 segundos.
- 2) Existe um atraso de 1 segundo entre desligar uma luz e ligar a próxima.
- 3) O semáforo da Major Road deve permanecer verde por pelo menos 30 segundos em cada ciclo, mas deve reagir ao sensor imediatamente após esse período.

B. Desenvolvimento e Apuramento de Resultados

1) Modelo:

O primeiro passo na resolução desta primeira parte consiste na modelação em UPPAAL do sistema de semáforos descrito anteriormente. Sendo assim, propomos o seguinte modelo:



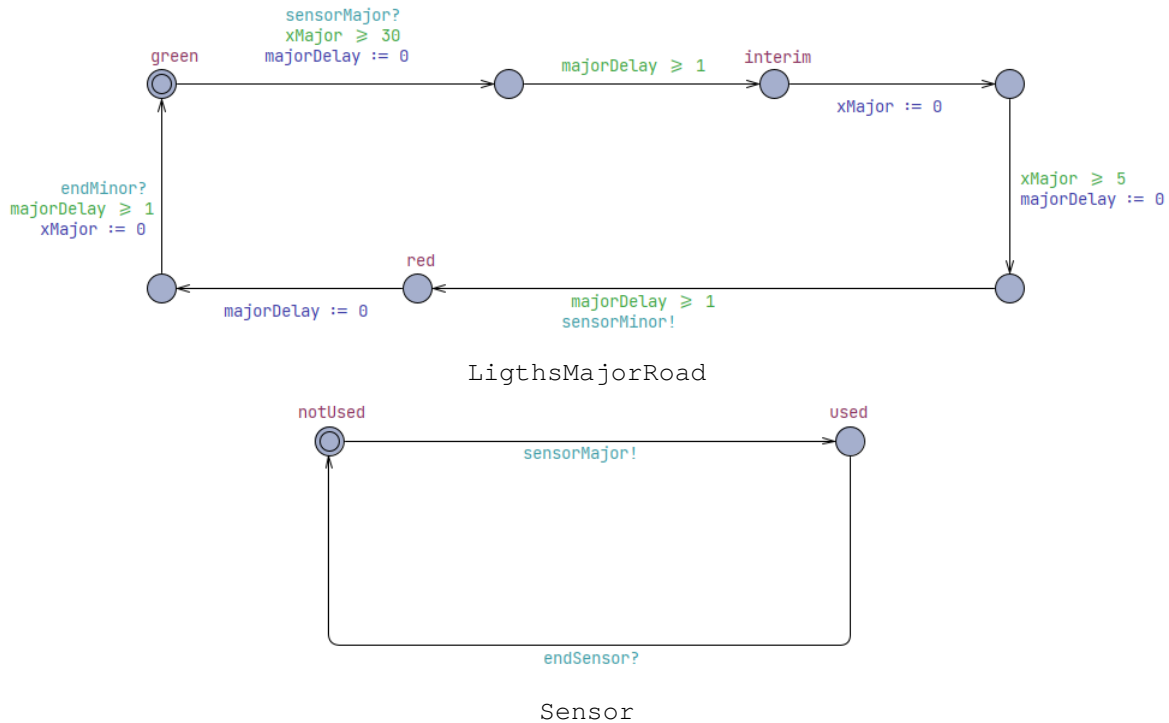


Fig. 2: Modelo desenvolvido

Como é possível ver na imagem acima, foram criadas três entidades (representadas sob a forma de autômatos) que ilustram, pela ordem em que aparecem, o comportamento dos semáforos na Minor Road, os semáforos na Major Road e o Sensor. Também estão declarados os diversos relógios que regulam o comportamento dos autômatos, sendo $xMinor$ e $minorDelay$ os relógios que regulam o tempo que cada luz fica acesa e o atraso na transição das mesmas na Minor Road, $xMajor$ e $majorDelay$ são o equivalente para a Major Road. Os canais de comunicação estabelecidos consistem em $sensorMinor$ e $endMinor$ que estabelecem comunicação entre a Major Road e a Minor Road, $sensorMajor$ que permite ao sensor comunicar com Major Road e $endSensor$ estabelecido entre Minor Road e o Sensor.

De seguida apresentamos uma descrição mais detalhada de cada autômato:

- **LightsMinorRoad:**

Este autômato é composto por três estados principais e quatro estados intermédios. Os estados principais consistem nas três possíveis luzes que um semáforo pode ter: vermelho (*red*), amarelo/intermédio (*interim*) e verde (*green*). Este autômato começa no estado inicial de vermelho, sendo apenas iniciado o ciclo de transição de luzes após ter recebido informação através do canal $sensorMinor$ enviada pelo semáforo da Major Road. O relógio $xMinor$ regula o tempo em que a luz verde fica acesa, precisamente 30 segundos, e a luz amarela, durante 5 segundos. O relógio $minorDelay$ garante que as transições entre cada luz só ocorrem durante 1 segundo. Quando está na transição entre a luz amarela e a luz vermelha, comunica com o sensor através de $endSensor$ permitindo que este volte ao estado de desativado. Após o estado intermédio, comunica com o semáforo de Major Road através de $endMinor$, permitindo que este acenda novamente as luzes verdes.

- **LightsMajorRoad:**

Este autômato é semelhante ao anterior, tendo os mesmos estados principais (*red*, *interim*, *green*) e os mesmos estados intermédios. A diferença está nas comunicações efetuadas e relógios usados. Contrariamente ao anterior, este autômato começa com um estado inicial de luz verde, sendo que o ciclo de luzes apenas começa após ser recebida informação da entidade Sensor pelo canal $sensorMajor$. O ciclo decorre da mesma forma que o semáforo da Minor Road, onde os relógios $xMajor$ e $majorDelay$ regulam o tempo em que as luzes verde e amarela ficam acesas e o tempo de transição entre cada luz, respetivamente. No instante antes de chegar ao estado de vermelho, utiliza o canal de comunicação $sensorMinor$ para indicar ao semáforo da Minor Road que é seguro iniciar a sua transição para a luz verde. Assim consegue-se garantir que as luzes dos semáforos nas duas estradas nunca são verde ou vermelho simultaneamente.

- **Sensor:**

Este autômato é o mais simples de todos e é composto por apenas dois estados, desativado (*notUsed*) e ativado (*used*). Ao ser ativado, envia um sinal aos semáforos da Major Road pelo canal *sensorMajor* de forma a que estes iniciem o seu ciclo de luzes, permitindo assim a gradual passagem de verde a vermelho. Após receber o sinal do semáforo da Minor Road através do canal *endSensor*, retorna ao estado de desativado.

O modelo foi devidamente testado para garantir um funcionamento dentro do esperado e que cumpra as regras estabelecidas. O anexo A ilustra a simulação feita em UPPAAL.

2) Verificação de Propriedades:

De forma a garantir que o modelo do sistema está bem construído e que as suas regras e restrições são respeitadas, expressamos em CTL as seguintes propriedades de *reachability*, *safety* e *liveness* e testamos-las no UPPAAL para verificar se eram satisfazíveis.

- (1) o semáforo da Minor Road pode ficar verde;
- (2) o semáforo da Major Road pode ficar vermelho;
- (3) o sistema nunca entra em estado de bloqueio (*deadlock*);
- (4) os semáforos da Minor e Major Road não podem estar verdes ao mesmo tempo;
- (5) se houver carros à espera, eventualmente eles irão ter o sinal verde.

Em CTL, as propriedades são expressas da seguinte forma:

- (1) `E<> LighthsMinorRoad.green`
- (2) `E<> LighthsMajorRoad.red`
- (3) `A[] not deadlock`
- (4) `A[] !(LighthsMajorRoad.green and LighthsMinorRoad.green)`
- (5) `Sensor.used and LighthsMinorRoad.red \leadsto LighthsMinorRoad.green`

Usando a ferramenta Verifier do UPPAAL (ver anexo B), testamos a satisfatibilidade de todas as propriedades tendo obtido resultado positivo em quatro dos cinco casos, sendo a quinta propriedade aquela que falha. Observando a simulação dos nossos autômatos, conseguimos perceber que uma das razões pela qual a quinta propriedade falha durante os testes de satisfatibilidade poderá ser o facto de os estados dos semáforos, tanto em Major Road com em Minor Road, terem a possibilidade de transitarem de estado mas de nunca serem obrigados ou forçados a fazê-lo. Infelizmente, não conseguimos contornar este problema, pelo que optamos por manter os autômatos no estado atual em que são apresentados, permitindo assim ter uma simulação sem ocorrências de *deadlocks*.

No decorrer do desenvolvimento deste trabalho, encontramos outras propriedades que consideramos desejáveis para o sistema ciber-físico em estudo, tais como:

- (6) é possível o semáforo da Major Road estar vermelho ao mesmo tempo que o semáforo da Minor Road está verde;
- (7) o semáforo da Minor Road fica sempre amarelo após terem passado 30 segundos de luz verde.

Usando a sintaxe em CTL abaixo, estas propriedades foram testadas no Verifier do UPPAAL (ver anexo B) tendo-se obtido o resultado de satisfazível nos dois casos:

- (6) `E<> LighthsMajorRoad.red and LighthsMinorRoad.green`
- (7) `A[] LighthsMinorRoad.interim imply xMinor > 30`

III. PARTE 2

A. Contextualização

O sistema anterior de semáforos funciona razoavelmente bem sob a suposição de que uma das estradas tem mais tráfego do que a outra. No entanto, essa suposição muitas vezes é muito forte: pode ser o caso de que ambas as estradas tenham a mesma quantidade de tráfego, ou até mesmo que o fluxo de tráfego varie drasticamente ao longo do dia.

A segunda parte deste trabalho (mais exploratório) tem como objetivo abordar precisamente esse problema, que é conhecido por ter um impacto significativo na economia e no meio ambiente. Para isso, agora podemos assumir que cada semáforo tem um sensor inteligente ligado a ele. O sensor informa se o tráfego próximo ao semáforo é alto, baixo ou simplesmente inexistente. Algumas das restrições da primeira parte mantêm-se.

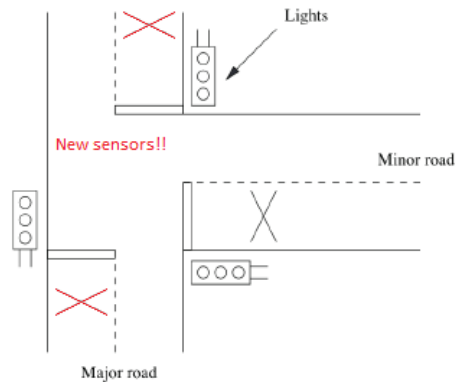


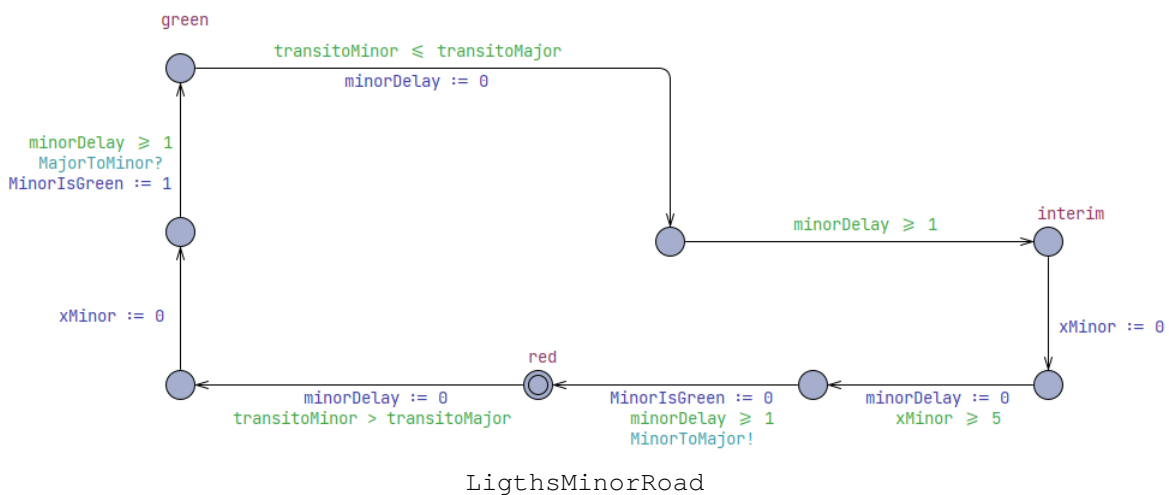
Fig. 3: Novo cenário em estudo

B. Desenvolvimento e Apuramento de Resultados

1) Modelo:

Para a resolução da segunda parte, foi necessário adaptar o modelo UPPAAL anterior para levar em consideração as informações fornecidas pelos sensores. Espera-se, por exemplo, que se o sensor mais à direita indicar tráfego intenso e os outros sensores não indicarem tráfego, então o semáforo mais à direita deve permanecer verde pelo menos até que os sensores forneçam novas informações. Sendo assim, propomos a seguinte reformulação para o modelo:

```
system LighsMinorRoad, LighsMajorRoad, SensorMinor, SensorMajor;
chan MinorToMajor, MajorToMinor;
clock xMinor, xMajor, minorDelay, majorDelay;
int transitoMinor;
int transitoMajor;
int MinorIsGreen;
int MajorIsGreen := 1;
```



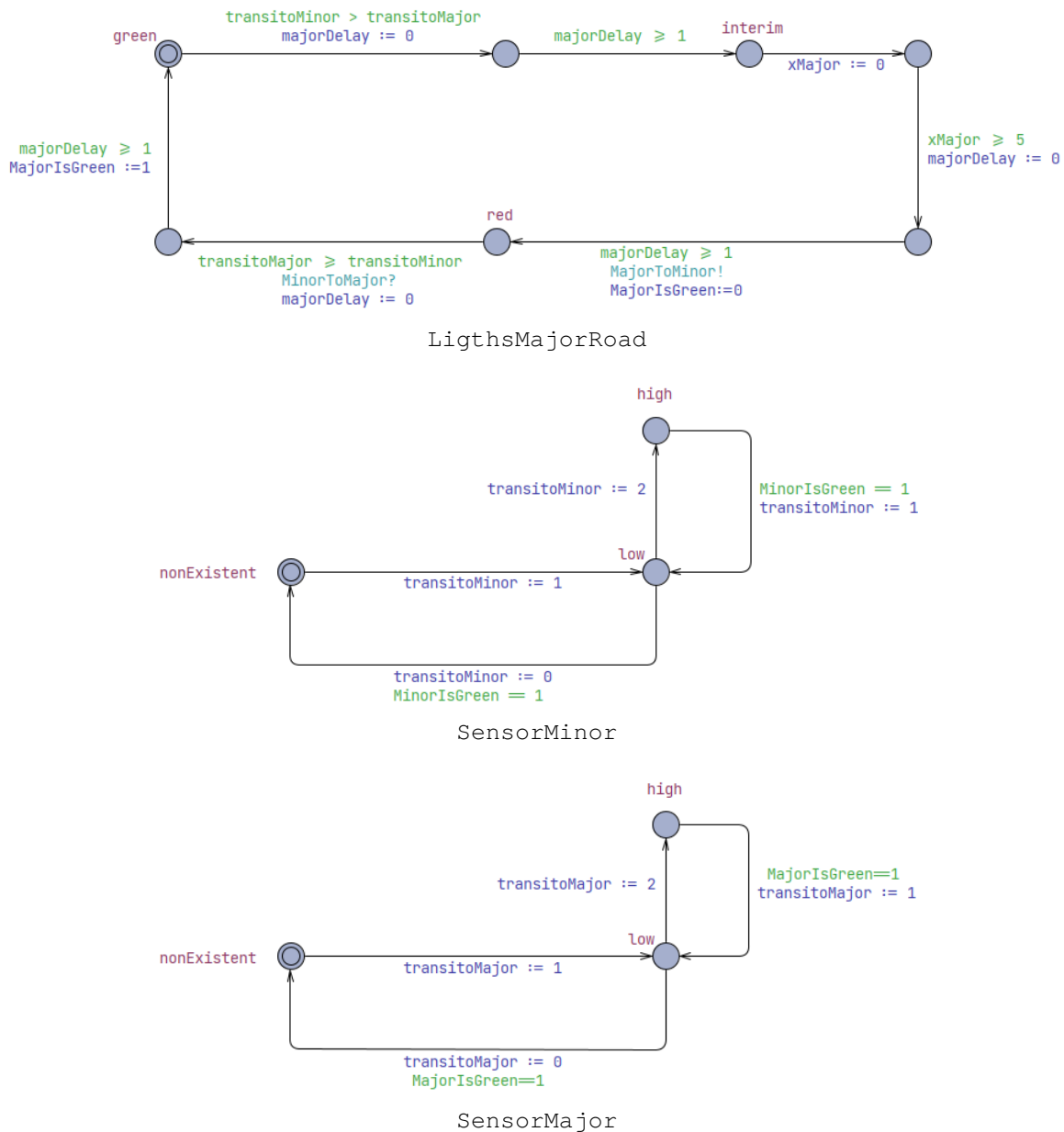


Fig. 4: Novo modelo desenvolvido

Relativamente ao modelo da primeira parte, foi adicionada uma quarta entidade que representa os dois sensores adicionados à estrada principal (Major Road). Os restantes três autómatos foram apenas modificados. Inicialmente, o plano era adaptar os autómatos ligeiramente de forma a tentar emular o novo cenário, através da adição de novos estados nos sensores, trocando os estados originais "notUsed" e "used" pelos três níveis de transito, "nonExistent", "low" e "high" do novo cenário e adicionando novos canais de comunicação. Assim, conseguiu-se simular o cenário, mas de forma muito limitada, devido a ocorrerem casos onde a Minor Road tinha mais trânsito e no entanto os seus semáforos não mudavam para verde até que a Major Road escoasse todo o seu trânsito, ou seja até atingir o nível "nonExistent". Isto ocorria devido ao facto de não existir um conjunto de variáveis que garantissem que a hierarquia entre os níveis de trânsito fosse respeitada, sendo essa hierarquia a seguinte:

$$high > low > nonExistent$$

A segunda versão do novo modelo foi feita a pensar nessa hierarquia, criando para esse efeito as variáveis *transitoMinor* e *transitoMajor* que são valores inteiros representativos do nível de trânsito detetado pelos sensores em cada estrada. Estas

variáveis tornam possível e fácil a comparação entre dois níveis de trânsito, passando esse a ser o critério para a troca das luzes nos semáforos, cumprindo assim a regra de que a estrada com maior nível de trânsito deve ter a luz verde. Nos casos onde os valores nas duas variáveis são iguais, decidimos que a Major Road deveria ter a vantagem sobre a Minor Road. A atribuição de valores a cada estado segue em baixo:

$$\begin{aligned} high &= 2 \\ low &= 1 \\ nonExistent &= 0 \end{aligned}$$

Também foram adicionadas as variáveis *MinorIsGreen* e *MajorIsGreen*. Estas variáveis atuam como valores "booleanos", alternando entre 0 e 1, e servem o propósito de indicar ao sensor se a luz no semáforo está efetivamente verde ou amarela, de forma a impedir que o nível de trânsito numa estrada transite de níveis hierarquicamente superiores para inferiores quando a luz está vermelha.

Também foram mantidas algumas das regras do cenário da parte 1 bem como algumas novas. Sendo assim, neste novo cenário o grupo teve em conta as seguintes restrições:

- 1) Os semáforos na Major Road iniciam com cor verde.
- 2) Os semáforos na Minor Road iniciam com cor vermelha.
- 3) Os semáforos intermediários (com luz amarela) permanecem ligados por 5 segundos.
- 4) Existe um atraso de 1 segundo entre desligar uma luz e ligar a próxima.

De seguida apresentamos uma descrição mais detalhada de cada autómato, dando particular atenção às modificações feitas em cada um:

• **LightsMinorRoad:**

Este autómato é composto por 3 estados principais e 5 estados intermédios. Os estados principais consistem nas três possíveis luzes que um semáforo pode ter: vermelho (red), amarelo/intermédio (interim) e verde (green). Da mesma forma que na primeira parte, o estado inicial é o vermelho visto que pretendemos manter uma hierarquia entre as estradas dando prioridade à Major Road. Assim, apenas pretendemos que este transite para o estado green quando o trânsito for superior ao trânsito da Major Road. Este ponto é verificado através das variáveis **transitoMinor** e **transitoMajor** atualizadas e controladas pelos sensores de ambas as estradas.

O relógio **minorDelay** mantém a sua função de controlar os segundos que são necessários para trocar as luzes e o relógio **xMinor** regula os 5 segundos que deve ficar em amarelo antes de transitar. Neste autómato também temos os canais de comunicação **MajorToMinor** e **MinorToMajor** que servem para as luzes de ambas as estradas comunicarem entre si de forma a trocarmos para verde apenas quando a outra já se encontra a vermelho. Também temos a variável **MinorIsGreen** que indica ao sensor quando é que o trânsito está a fluir, de forma a que entre verde e vermelho o trânsito possa diminuir, visto que é possível a circulação de carros.

• **LightsMajorRoad:**

Este autómato é composto por 3 estados principais e 4 intermédios. Os estados principais consistem nas três possíveis luzes que um semáforo pode ter: vermelho (red), amarelo/intermédio (interim) e verde (green). O estado inicial é o green para, como já dito anteriormente, atribuir uma prioridade à Major Road. Assim apenas pretendemos que transite para vermelho quando a Minor Road apresentar trânsito superior; se for igual ou inferior a Major Road mantém o estado green. Isto é verificado através das variáveis **transitoMinor** e **transitoMajor**.

O relógio **majorDelay** mantém a sua função de controlar os segundos que são necessários para trocar as luzes e o relógio **xMajor** regula os 5 segundos que deve ficar em amarelo antes de transitar. Neste autómato também temos os canais de comunicação **MajorToMinor** e **MinorToMajor** que servem para as luzes de ambas as estradas comunicarem entre si de forma a trocarmos para verde apenas quando a outra já se encontra a vermelho. Além disto, o autómato apresenta a variável **MajorIsGreen** inicializada a 1 visto que começa a green, permitindo assim a circulação do trânsito desde o estado inicial e atualizada quando transita para vermelho de forma a começar acumular o trânsito.

• **SensorMinor:**

Este autómato é mais simples e retrata o sensor da Minor Road. É composto por 3 estados principais: nonExistent (sem trânsito), low (pouco trânsito) e high (muito trânsito). Neste autómato, é atualizada a variável **transitoMinor** responsável pela indicação do nível de trânsito da estrada e serve como ponto de comparação para gerir as prioridades. O estado inicial

é `nonExistent` assumindo que não há trânsito inicialmente. De notar que a variável atualizada nos autómatos anteriores **MinorIsGreen** serve de guarda para as transições de diminuição de trânsito.

- **SensorMajor:**

Este autómato é igual ao anterior apenas com os nomes das variáveis referentes à Major Road, de forma a servir o propósito de regular o trânsito na Major Road.

A ilustração da simulação destes autómatos no UPPAAL encontra-se no anexo C.

2) *Verificação de Propriedades:*

Iremos agora verificar se todas as propriedades mencionadas na primeira parte ainda são válidas para o novo modelo. Devido a algumas limitações na ferramenta Verifier do UPPAAL, a quinta propriedade será dividida em duas, ficando assim:

- (5.1) se houver carros à espera na Major Road, eventualmente eles irão ter o sinal verde.
- (5.2) se houver carros à espera na Minor Road, eventualmente eles irão ter o sinal verde.

Convertendo 5.1 e 5.2 em CTL, ficamos com:

- (5.1) $(\text{SensorMajor.low or SensorMajor.high}) \text{ and LightsMajorRoad.red} \rightsquigarrow \text{LightsMajorRoad.green}$
- (5.2) $(\text{SensorMinor.low or SensorMinor.high}) \text{ and LightsMinorRoad.red} \rightsquigarrow \text{LightsMinorRoad.green}$

Fazendo os devidos testes, obtemos o mesmo resultado que na parte 1, onde as quatro primeiras propriedades obtiveram o resultado de satisfazível, ficando 5.1 e 5.2 a falhar. A razão poderá ser a mesma que foi mencionada na parte 1, onde os autómatos podem transitar de estado mas não são obrigados a isso. Ou seja, mesmo que no decorrer da simulação do modelo seja possível observar que se houver carros à espera, eventualmente eles recebem a luz verde, isso não significa que o modelo esteja a ser obrigado a fazer essas transições que visualizamos. No anexo D pode encontrar-se os resultados da verificação no UPPAAL.

3) *Valorização:*

Para concluir a exploração deste novo modelo de natureza mais exploratória, e para efeitos de valorização, propomos as seguintes propriedades expressas em CTL que digam algo sobre a eficiência do sistema desenvolvido. Escolhemos as seguintes propriedades:

- (8) Se houver mais trânsito na Minor Road, então o seu semáforo eventualmente será o único que está verde
- (9) Se houver mais trânsito na Major Road, então o seu semáforo eventualmente será o único que está verde

Em CTL, expressamos as propriedades 8 e 9 como:

- (8) $A <> (\text{transitoMinor} > \text{transitoMajor}) \text{ imply } (\text{LightsMinorRoad.green and LightsMajorRoad.red})$
- (9) $A <> (\text{transitoMinor} \leq \text{transitoMajor}) \text{ imply } (\text{LightsMinorRoad.red and LightsMajorRoad.green})$

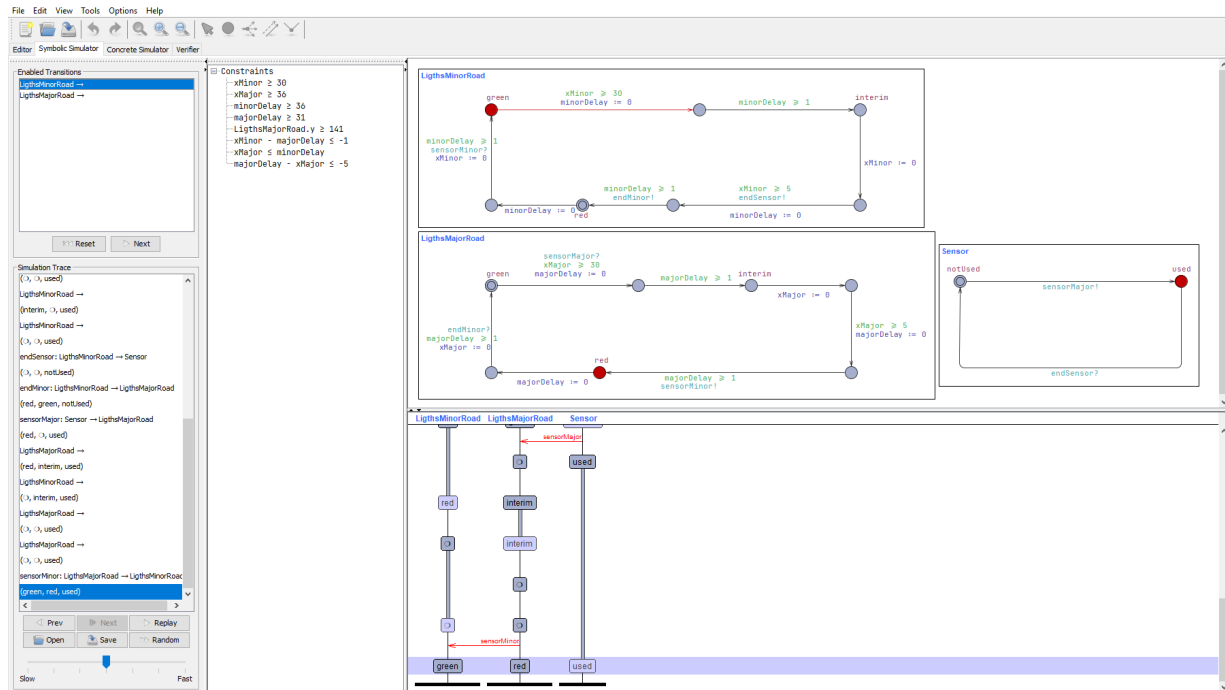
Ambas obtiveram o resultado de satisfazível.

IV. CONCLUSÃO

Com este projeto terminado, apresentamos uma breve conclusão que consideramos englobar todo o processo de aprendizagem despoletado por este trabalho. Em suma, consideramos que o trabalho desenvolvido e os resultados apurados são positivos, demonstrando uma clara capacidade em aplicar os conceitos das aulas de Programação Ciber-Física. Destacamos como pontos positivos as soluções encontradas para cada um dos cenários e em particular o cumprimento das regras de cada um. No entanto, reconhecemos que existem alguns pontos a melhorar, tais como aumentar a complexidade de alguns dos autómatos, permitindo assim que as propriedades que não se conseguiram provar nas duas fases exploradas passassem a ser satisfazíveis. Como trabalho futuro, seria também interessante introduzir cenários ainda mais complexos, que envolvessem, por exemplo, mais do que duas estradas, introduzindo assim um novo conjunto de desafios.

APPENDIX A

SIMULAÇÃO DO UPPAAL NA PARTE 1



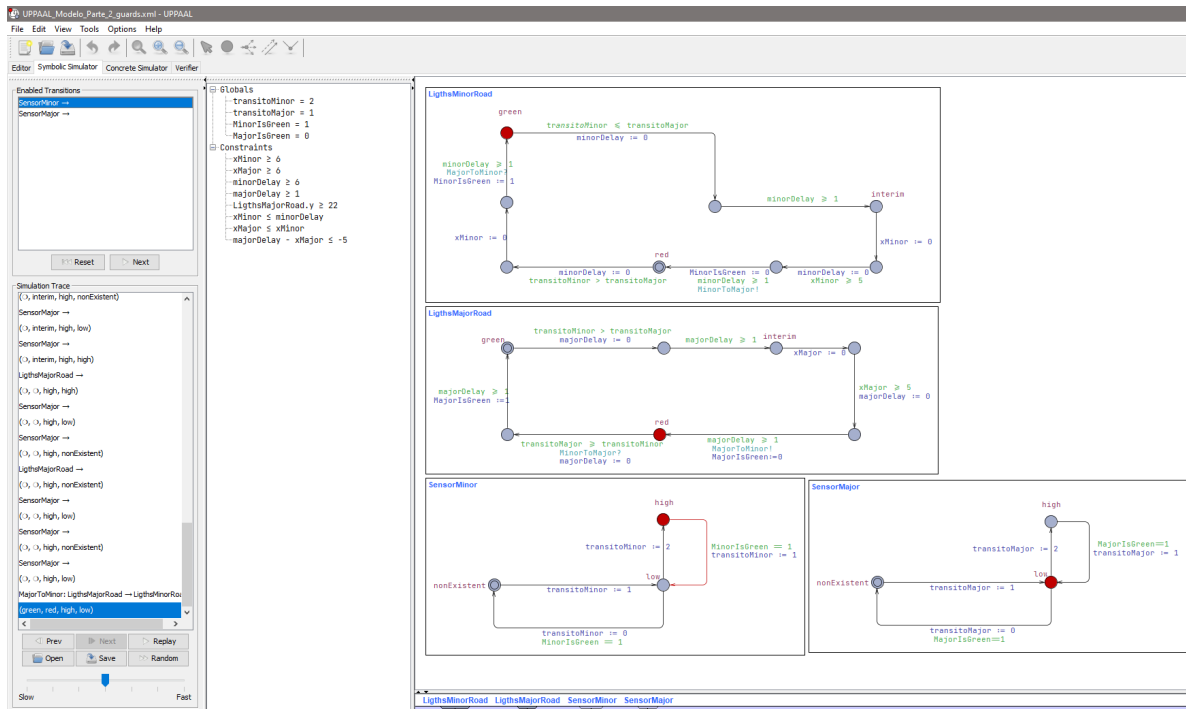
APPENDIX B

RESULTADOS VERIFIER DO UPPAAL NA PARTE 1

Editor	Symbolic Simulator	Concrete Simulator	Verifier
Overview			
E◊ LigthsMinorRoad.green			●
E◊ LigthsMajorRoad.red			●
A[] not deadlock			●
A[] !(LigthsMajorRoad.green and LigthsMinorRoad.green)			●
Sensor.used and LigthsMinorRoad.red → LigthsMinorRoad.green			●
E◊ LigthsMajorRoad.red and LigthsMinorRoad.green			●
A[] LigthsMinorRoad.interim imply xMinor > 30			●

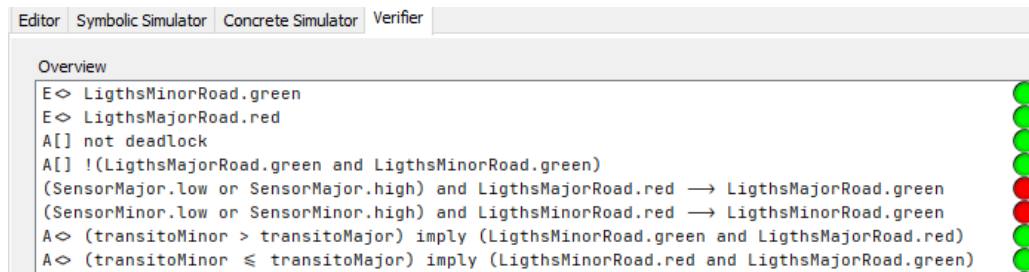
APPENDIX C

SIMULAÇÃO DO UPPAAL NA PARTE 2



APPENDIX D

RESULTADOS VERIFIER DO UPPAAL NA PARTE 2



REFERENCES

- [1] <https://uppaal.org/features/>
- [2] <https://haslab.github.io/MFP/PCF/2324/index>