

Trabalho Prático 1 - PCF

Jéssica Fernandes (A93318) Vitor Lelis (PG54273)

April 07, 2024

Contents

1. Introdução	3
1.1. Restrições	3
1.2. Condições de verificação	4
2. Primeira Parte	4
2.1. Modelação	4
2.1.1. Sensor	4
2.1.2. Semáforo da Estrada Principal	5
2.1.3. Semáforo da Estrada Secundária	6
2.2. Verificação	6
3. Segunda Parte	7
3.1. Descrição	7
3.2. Modelação	7
3.2.1. Sensores	7
3.2.2. Controller	8
3.2.3. Semáforos	9
3.3. Verificação	10
4. Conclusão	10
Bibliography	11
Index of Figures	12

1. Introdução

Este relatório tem como objetivo detalhar o processo de modelação e análise de um Sistema Ciber-Físico em UPPAAL [1], concebido no âmbito do primeiro trabalho prático da unidade curricular “Programação Ciber Física”.

A situação em análise consiste num par de ruas perpendiculares, que dispõem de um par de semáforos. O objetivo do trabalho é conceber um sistema que garanta o fluxo correto dos carros em ambas as ruas, permitindo a sua circulação alternada, gerindo o nível de tráfego, e evitando possíveis colisões.

O trabalho encontra-se dividido em dois cenários: no primeiro, devemos abordar a situação considerando a presença de apenas um sensor na estrada mais pequena, o qual alerta para a presença de carros a circular na mesma. No segundo, cada estrada possui um destes sensores, que, neste cenário, monitorizam o nível de tráfego. Devemos desenvolver um sistema capaz de distinguir quando permitir a passagem numa ou noutra estrada, pela análise do nível de fluxo em ambas, a cada momento.

Começaremos por indicar as restrições que devem ser cumpridas, para garantir que as soluções que propomos satisfazem as necessidades do sistema, em cada um dos cenários. De seguida, enumeraremos as condições de verificação utilizadas para verificar a correção dos programas.

Posteriormente, passaremos a apresentar as propostas de soluções desenvolvidas, incluindo uma explicação dos modelos e das escolhas aquando do seu desenvolvimento, bem como os testes que realizamos e as conclusões retiradas.

1.1. Restrições

Para o primeiro cenário, foram estabelecidas as seguintes restrições:

1. A posição padrão dos semáforos da estrada principal é verde, e na estrada secundária para vermelho, a menos que um veículo seja detetado pelo sensor;
2. Quando um veículo é detetado pelo sensor, os semáforos mudarão, permitindo o tráfego sair da via secundária, a menos que o tempo mínimo (30 segundos) não tenha passado ainda. Neste caso, o semáforo da rua principal continuará verde, trocando imediatamente após o intervalo de tempo mínimo ter sido cumprido.
3. O semáforo da rua secundária permanece verde sempre apenas durante 30 segundos, voltando à sua posição padrão após este intervalo de tempo, permitindo de novo o fluxo na rua principal;
4. Assim que um veículo for detetado pelo sensor, este será desativado até que os semáforos da via secundária voltem a ficar vermelhos, pois não queremos bloquear a longo prazo o fluxo na rua principal;
5. A luz amarela dos semáforos permanece ligada por 5 segundos, para permitir aos carros abrandarem antes de passar a vermelho;
6. Existe um intervalo de 1 segundo entre um semáforo ficar vermelho, e o outro ficar verde, para garantir que não estejam dois verdes ao mesmo tempo, evitando colisões;

Para o segundo cenário, as condições são diferentes, e assim são também as restrições:

1. Não existe uma posição padrão neste cenário, é arbitrário.

2. A luz de cada semáforo é determinada pela comparação de fluxos de carros nos sensores dos dois semáforos;
3. Se um sensor deteta fluxo alto e o outro deteta que não existe fluxo, a luz do primeiro deve estar verde.
4. Se um sensor deteta fluxo alto e o outro deteta que um fluxo baixo, a luz do primeiro deve estar verde.
5. A luz amarela dos semáforos permanece ligada por 5 segundos, para permitir aos carros abrandarem antes de passar a vermelho;
6. Existe um intervalo de 1 segundo entre um semáforo ficar vermelho, e o outro ficar verde, para garantir que não estejam dois verdes ao mesmo tempo, evitando colisões;

1.2. Condições de verificação

Para garantir a precisão dos modelos, foram propostas algumas verificações, que devem ser cumpridas em **ambos** os cenários São elas:

- (V1) O semáforo da rua secundária **deve ser** capaz de ficar verde;
- (V2) O semáforo da rua secundária **deve ser** capaz de ficar vermelho;
- (V3) Os semáforos **não podem** ficar verdes simultaneamente;
- (V4) Se houver carros à espera, eles **devem eventualmente** receber a luz verde.
- (V5) O sistema **não deve** entrar em *deadlock*;

Estas condições foram traduzidas para a sintaxe do programa e testadas no Verifier, da seguinte forma:

- (V1) `E<> (SecondaryTrafficLight.GREEN)`
- (V2) `E<> (MainTrafficLight.RED)`
- (V3) `A[] !(MainTrafficLight.GREEN && SecondaryTrafficLight.GREEN)`
- (V4) `(Sensor.OFF && SecondaryTrafficLight.RED && !MainTrafficLight.RED) -> SecondaryTrafficLight.GREEN`
- (V5) `A[] not deadlock`

2. Primeira Parte

2.1. Modelação

Foram concebidos 3 componentes para esta solução: os dois semáforos e o sensor que gere os seus estados.

2.1.1. Sensor

O sensor foi o ponto de partida, uma vez que é o componente que tem completo controlo sobre qual semáforo deve estar verde, pois deteta a passagem dos carros. Começamos por um template simples com apenas dois estados: **ON** e **OFF** que representam se o sensor se encontra ligado ou desligado, sendo o estado inicial o **ON**, pois é o estado correspondente ao semáforo da estrada secundária estar vermelho, que é o estado padrão.

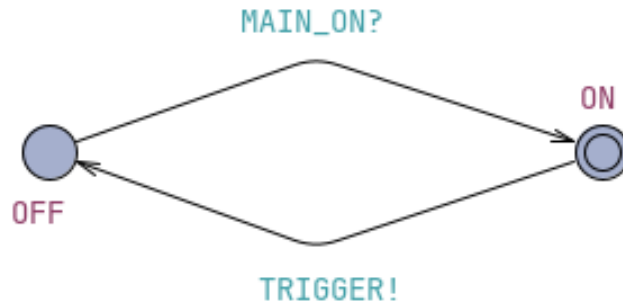


Figure 1: Sensor da Primeira Parte

O sensor muda de estado de **OFF** para **ON** quando a estrada principal tem o semáforo verde (**MAIN_ON**), pois é o momento de voltar a detetar a passagem dos carros. Quando passa para o estado de desligado (**OFF**), por ter sido ativado por um carro, envia um sinal de **TRIGGER** que avisa o semáforo da estrada principal que deve ficar vermelho, para permitir o tráfego na estrada secundária.

2.1.2. Semáforo da Estrada Principal

De seguida, temos o componente do semáforo da estrada principal. Este semáforo possui 3 estados: **GREEN**, **YELLOW** e **RED**.

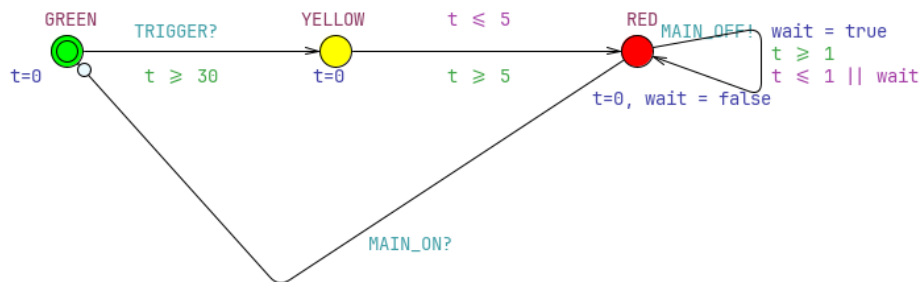


Figure 2: Semáforo Principal

O estado inicial do semáforo principal é verde, pois é o seu estado padrão. A passagem de estado do **GREEN** para **YELLOW** é ativada pelo **TRIGGER** que mencionamos no sensor, assim que um carro é detetado. Há, porém, uma condição a cumprir - terem passado 30 segundos, que é o intervalo mínimo, segundo as restrições ($t \geq 30$).

Do estado **YELLOW** para **RED** possuímos uma guarda que garante que passaram 5 segundos obrigatoriamente, devido também à restrição das luzes amarelas, e, adicionalmente, um invariante, que garante que, entre os estados mencionados, o tempo passado está entre 0 e 5 segundos. Isto vai garantir, em suma, que esta passagem dura 5 segundos.

O estado **RED** possui um trigger **MAIN_OFF** que corresponde ao aviso de que o semáforo da rua principal está agora vermelho. Permanece neste estado durante 1 segundo ($t \geq 1$) devido ao *delay* que deve existir entre a mudança de um semáforo para vermelho, e do outro para verde. Desta forma, garantimos que os dois semáforos nunca estão verdes simultaneamente, e para assegurar isso, possuímos ainda um **wait** que começa com o valor *true* no estado **RED** e permanece assim até ao começo da transição seguinte onde se torna *false*.

O semáforo passa de **RED** a **GREEN** quando recebe a informação de que pode voltar a ficar verde (**MAIN_ON**), não havendo restrições de tempo nesta mudança de estado.

2.1.3. Semáforo da Estrada Secundária

Por fim, modelamos o semáforo da estrada secundária. Os seus estados correspondem exatamente aos do semáforo da estrada principal, porém, o seu estado inicial é o **RED**.

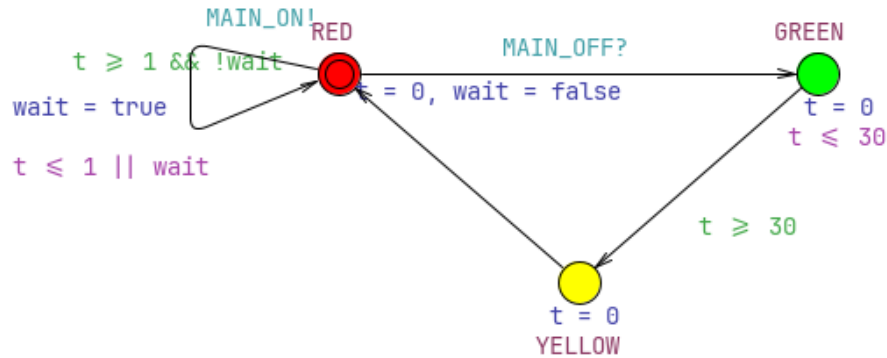


Figure 3: Semáforo Secundário

Este semáforo passa ao estado **GREEN**, assim que o semáforo da estrada principal informa que vai passar a verde (**MAIN_OFF**), não havendo restrições de tempo nesta passagem.

Como indicado nas restrições, fica verde durante 30 segundos, e, de seguida, passa ao estado amarelo. Existe, portanto, uma guarda $t \geq 30$ para garantir que passaram efetivamente 30 segundos, antes da mudança para o amarelo (**YELLOW**), e, um invariante, que garante que o semáforo passa, no máximo, 30 segundos no estado verde (**GREEN**).

A passagem do amarelo (**YELLOW**) para o vermelho (**RED**) acontece exatamente da mesma forma do semáforo da rua principal. A parte do sistema que garante o 1 segundo no estado vermelho, como mencionado anteriormente, funciona também de forma análoga ao semáforo da estrada principal, mas, desta vez, possuímos **MAIN_ON** em vez de **MAIN_OFF**.

2.2. Verificação

Para verificar a correção deste modelo, utilizamos as condições de verificação já enunciadas no subcapítulo 1.2.

Todas as verificações passaram, garantindo, assim, que o programa funciona como se pretende.

```
(Sensor_OFF && SecondaryTrafficLight.RED && !MainTrafficLight.RED) → SecondaryTrafficLight.GREEN
A[] not deadlock
E◊ (SecondaryTrafficLight.GREEN)
E◊ (MainTrafficLight.RED)
A[] !(MainTrafficLight.GREEN && SecondaryTrafficLight.GREEN)
```

Figure 4: Verificação

Também utilizamos o Symbolic Simulator para visualizar melhor as transições entre estados, e garantir que o modelo se comportava como pretendido, fazendo a passagem entre estados corretamente, e sincronizados de forma coerente entre os vários componentes.

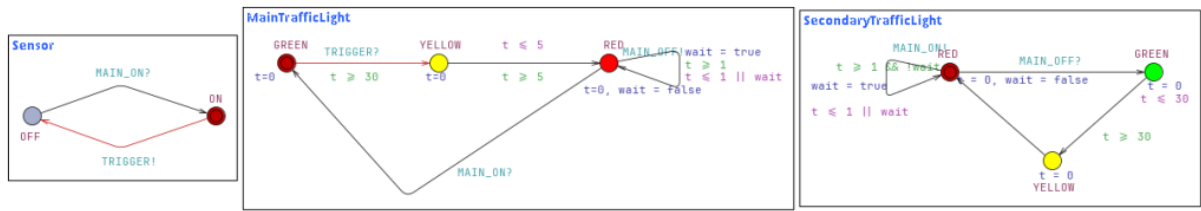


Figure 5: Componentes

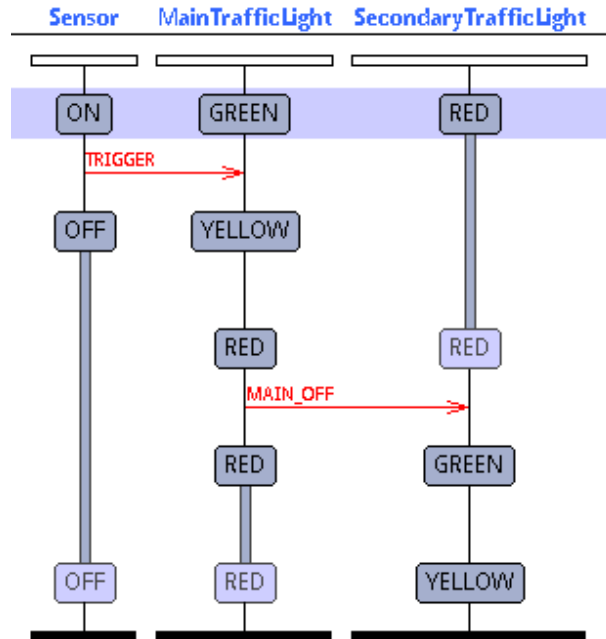


Figure 6: Transições

3. Segunda Parte

3.1. Descrição

Na segunda parte, foi sugerido considerar o cenário com dois sensores para cada uma das estradas, visando a sincronização entre eles, levando ainda em conta as restrições mencionadas anteriormente.

Além disso, foi assumida também uma categorização da quantidade de tráfego até os semáforos, dividindo-os em: **High**, **Low** e **Non existent**.

3.2. Modelação

Para a modelação desta fase foram feitas algumas alterações no sistema anterior para estarem de acordo com o problema. Neste cenário, a luz de cada semáforo é determinada pela comparação de fluxos de carros nos sensores dos dois semáforos. Assim sendo, criaram-se dois novos componentes: o novo **Sensor do Semáforo Principal**, com os devidos estados **high**, **low**, **no** (non existent), e o **Controller**.

3.2.1. Sensores

Os Sensores funcionam de forma análoga entre si, diferindo apenas no nome dos seus canais. Os seus estados são correspondentes ao nível de tráfego na rua que monitorizam: **NO**, **LOW**

e **HIGH**. De cada vez que um estado muda, seja entre que estados for, existe um *trigger* que ativa essa mudança e a comunica ao controller pelo canal respetivo.

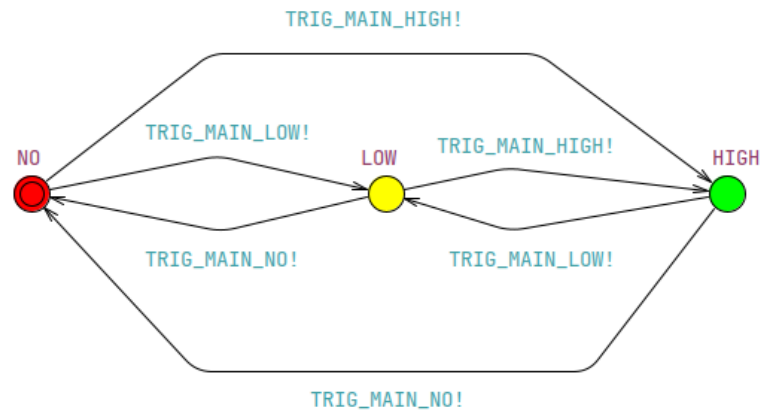


Figure 7: Sensor Principal da Segunda Parte

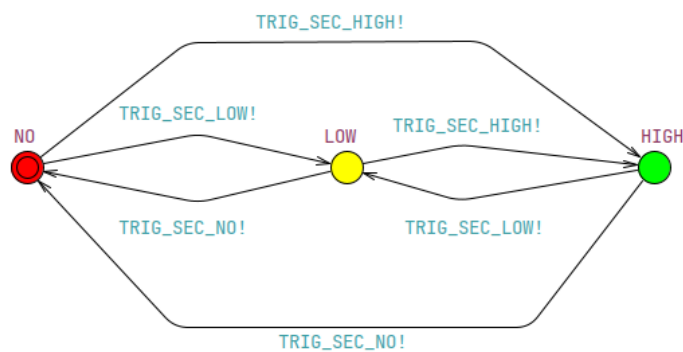


Figure 8: Sensor Secundário da Segunda Parte

3.2.2. Controller

O Controller, mencionado anteriormente, monitoriza as mudanças de estado dos sensores, que revelam o tráfego do semáforo pelo qual são responsáveis.

O Controller possui dois estados: O **MAIN_ON** correspondente ao cenário em que o semáforo da rua principal deve estar verde, e o **SEC_ON**, correspondente ao cenário em que o semáforo da rua secundária deve estar verde.

Sempre que um sensor deteta a mudança de estado de um semáforo, o Controller atualiza o seu estado. Quando um dos semáforos muda o seu estado para **low** ou **no** e o outro para **high**, ou vice versa, o controller identifica que uma mudança de semáforo verde deve ser feita, e ativa o **trigger** correspondente ao semáforo que deve estar verde, para ambos os lados.

3.3. Verificação

Como podemos ver na figura, a verificação V5 foi a única que não foi satisfeita. Infelizmente, não foi possível identificar o motivo deste problema atempadamente.

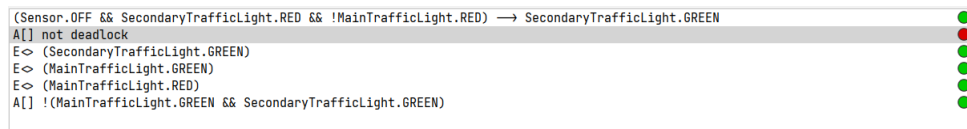


Figure 12: Verificação da Segunda Parte

4. Conclusão

Para a realização deste Trabalho Prático, foi proposto o desafio de modular o comportamento de dois semáforos em dois cenários diferentes utilizando a ferramenta UPPAAL [1]. A partir do fornecimento de condições e verificações formais, foram então criadas duas soluções para responder a estes problemas.

Na primeira parte, a desconstrução do problema e o desenho da solução foi relativamente simples, salvo alguns pormenores. O verdadeiro desafio encontrado foi desenvolver a solução utilizando a sintaxe exigida pela ferramenta que devíamos utilizar. Após alguma familiarização e várias tentativas, foi possível encontrar um modelo totalmente satisfatório às exigências do cenário em causa.

Na segunda parte, a maior dificuldade foi a complexidade de situações possíveis que devíamos ter em conta na construção da solução, bem como a sua compilação e organização num sistema que satisfizesse de forma ótima a gestão do tráfego em ambas as estradas. A ideia de existir um controller que monitora a mudança de estados facilita a logística, mas infelizmente, depois de várias tentativas, ainda surgiram alguns problemas, que fazem com que a solução funcione apenas parcialmente, gerando, por vezes, *deadlocks*.

Por fim, apesar das dificuldades encontradas, maioritariamente na segunda parte, o enunciado proposto foi desafiante mas gratificante. Apesar de reconhecer os aspetos ainda a melhorar, grupo considera-se satisfeito com o trabalho desenvolvido, as aprendizagens fundamentais adquiridas e o progresso feito na utilização desta ferramenta e sintaxe.

Bibliography

[1] “UPPAAL.” [Online]. Available: <https://uppaal.org/>^o

Index of Figures

Figure 1: Sensor da Primeira Parte	5
Figure 2: Semáforo Principal	5
Figure 3: Semáforo Secundário	6
Figure 4: Verificação	6
Figure 5: Componentes	7
Figure 6: Transições	7
Figure 7: Sensor Principal da Segunda Parte	8
Figure 8: Sensor Secundário da Segunda Parte	8
Figure 9: Controller da Segunda Parte	9
Figure 10: Semáforo Principal da Segunda Parte	9
Figure 11: Semáforo Secundário da Segunda Parte	9
Figure 12: Verificação da Segunda Parte	10