

Trabalho Prático 1

Arquitetura e Cálculo - 2019/2020

Bruno Antunes - PG40866

Pedro Moura - PG41094

7 de Maio de 2020

Resumo

Modelação e análise de um sistema de tempo-real com base na ferramenta UPPAAL.

Conteúdo

1	Introdução	2
2	Modelação	3
2.1	<i>Lantern</i>	3
2.2	<i>Bridge</i>	4
2.3	<i>Person</i>	5
3	Verificação	6
4	Conclusão	7

1 Introdução

O presente relatório descreve o desenvolvimento do 1º trabalho prático da UC de Arquitetura e Cálculo, integrada no perfil de Métodos Formais em Engenharia de *Software*, do 1º ano do Mestrado em Engenharia Informática da Universidade do Minho.

Este trabalho consiste na modelação e análise de um sistema de tempo-real através da ferramenta UPPAAL.

Enunciado *In the middle of the night, four adventurers encounter a shabby rope-bridge spanning a deep ravine. For safety reasons, they decide that no more than 2 people should cross the bridge at the same time and that a flashlight needs to be carried by one of them in every crossing. They have only one flashlight. The 4 adventurers are not equally skilled: crossing the bridge takes them 1, 2, 5, and 10 minutes, respectively. A pair of adventurers crosses the bridge in an amount of time equal to that of the slowest of the two adventurers. One of the adventurers claims that they cannot be all on the other side in less than 19 minutes. One companion disagrees and claims that it can be done in 17 minutes.*

Objetivo Criar um modelo para este sistema e verificar, através de CTL, as seguintes propriedades:

- é possível que todos os aventureiros estejam do outro lado da ponte em 17 minutos;
- é impossível que todos os aventureiros estejam no outro lado da ponte em menos de 17 minutos.

2 Modelação

No sistema existem 3 tipos de autómatos: **Person** ($\times 4$), **Lantern** e **Bridge**.

Para além disso, é necessária a existência de algumas variáveis globais:

- L_L - inteiro que representa a localização da lanterna. 0 se no início da ponte, 1 se no fim;
- L_BUSY - booleano que diz se a lanterna está ocupada (*true*), ou está livre (*false*). É inicializado a *false*.
- *global* - relógio global.

2.1 *Lantern*

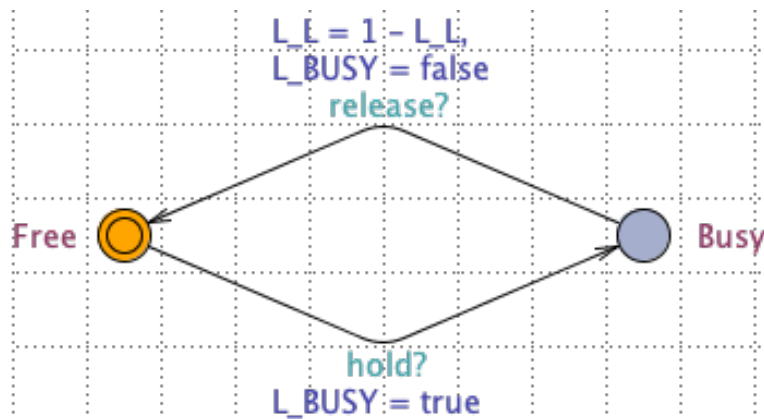


Figura 1: **Lantern** - modelo.

O processo **Lantern** representa a lanterna.

Quando sincroniza através da ação *hold?*, significando que alguém pegou na lanterna, avança para um estado em que está ocupada, e a partir daqui, só com a transição *release?* é que volta a estar livre. O modelo deste processo também faz com que seja impossível o sistema executar, por exemplo, 2 ações *hold!* seguidas, o que significa que apenas um aventureiro pode estar com a lanterna num certo momento.

De notar ainda que, como vamos ver a seguir, no fim de cada travessia, o aventureiro que possui a lanterna tem de a libertar. Com isto, podemos atualizar a posição da lanterna aquando de um *release?*.

2.2 *Bridge*

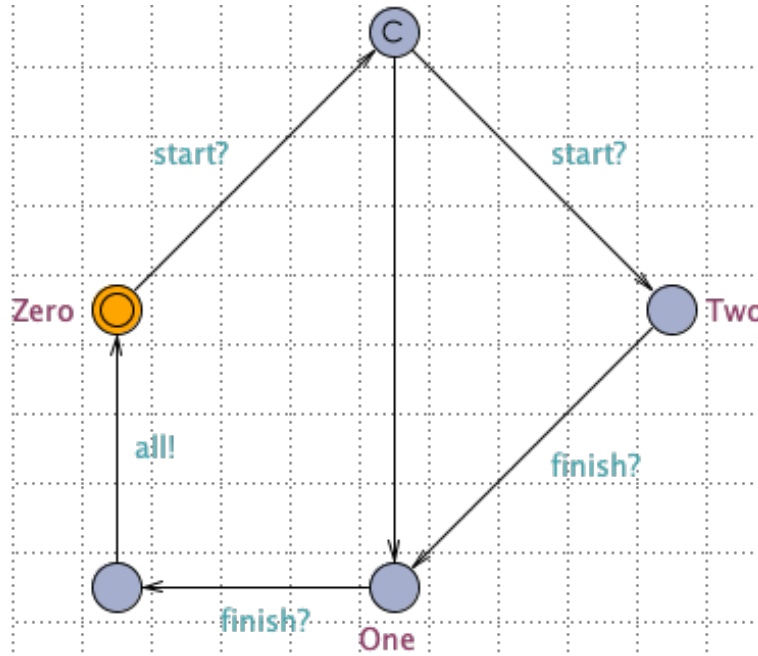


Figura 2: **Bridge** - modelo.

O processo **Bridge** controla o número de elementos que realizam uma travessia.

Como podemos ver, num primeiro momento o processo é avisado que alguém quer fazer uma travessia (**start?**), e obriga a que a próxima transição do sistema tem de ser uma das seguintes: ou outro aventureiro inicia a travessia (e a ponte passa a saber que a travessia vai ser realizada por 2 pessoas), ou evolui para um estado que significa que apenas 1 pessoa irá realizar a travessia. Em ambos os casos, reparamos que este processo não aceita mais nenhuma transição (**start!**), o que significa que o máximo de pessoas que realizar uma travessia ao mesmo tempo são apenas 2.

Dependendo de serem 1 ou 2 pessoas a realizar a travessia, o processo espera receber, respetivamente, 1 ou 2 notificações que os intervenientes acabaram a travessia (**finish?**). Depois disto, avisa o aventureiro que tem a lanterna que todos já acabaram (**all!**), voltando ao estado inicial e ficando pronto para acolher nova travessia,

2.3 *Person*

Processo que emula uma travessia de um aventureiro. Temos 3 variáveis locais a ter em conta:

- `delay` - inteiro que é recebido como parâmetro e indica o tempo que o aventureiro demora a atravessar a ponte.
- `L_P` - inteiro que representa a localização do aventureiro. 0 se no início da ponte, 1 se no fim. Começa a 0.
- `t` - relógio local.

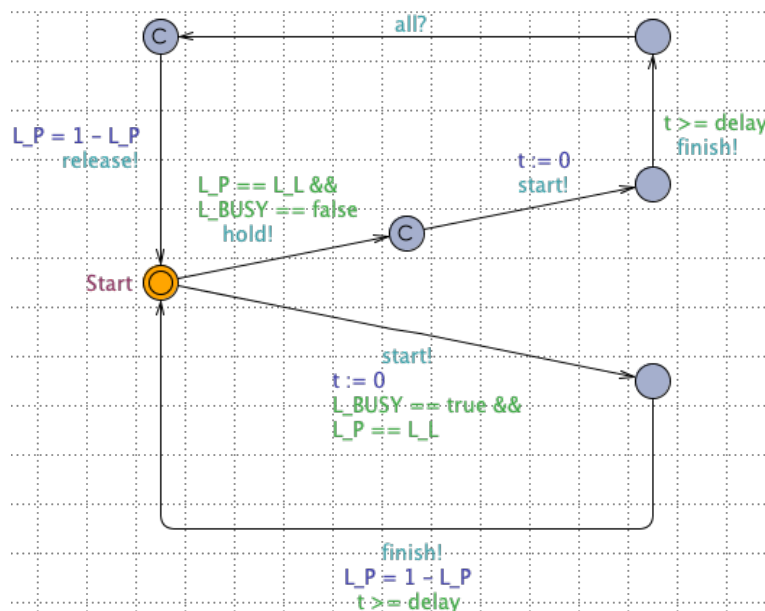


Figura 3: **Person** - modelo.

Um aventureiro pode realizar uma travessia de 2 formas:

- com lanterna: se a lanterna estiver livre e no mesmo local que o aventureiro, este pega na lanterna (**hold!**) e inicia imediatamente a travessia (**start!**). Neste ponto, pode esperar que outro aventureiro inicie também uma travessia, ou avança sozinho.

Em ambos os casos, depois de esperar *delay* minutos, diz ao processo **Bridge** que está pronto para acabar a travessia (**finish!**), e espera

que este lhe responda indicando se todos os intervenientes da travessia já a acabaram (**all?**), podendo assim libertar a lanterna, atualizar a sua posição e voltar ao estado inicial, de modo a poder iniciar nova travessia (ou não).

- sem lanterna: o aventureiro só consegue iniciar a travessia se já existir um outro aventureiro que tenha iniciado a sua travessia com a lanterna e do mesmo lado que ele. Depois de iniciada a travessia, o sistema espera *delay* minutos até poder acabar a sua travessia (**finish!**), voltando ao estado inicial com a sua posição atualizada.

Apesar de ter voltado ao estado inicial, este aventureiro não pode iniciar outra travessia enquanto que a lanterna esteja liberta. O que acontece apenas quando o aventureiro com a lanterna acaba a sua travessia, e este só acaba quando todos os todos os que fizeram esta travessia acabarem. Isto pode ser visto como "acabar ao mesmo tempo", porque mesmo que um execute a ação **finish!** primeiro que o outro, a travessia só realmente acaba quando ambos estão do outro lado da ponte e a lanterna é largada, sendo o tempo da travessia igual ao maior *delay* entre os intervenientes, como era pedido no exercício.

3 Verificação

Não existe *deadlock*: $A \Box \neg \text{deadlock}$

$A[] \text{ not deadlock}$

É possível que todos os aventureiros estejam do outro lado da ponte em 17 minutos: $E \Diamond (\forall_{1 \leq i \leq 4} \text{Person}_i.L_P = 1) \wedge \text{global} = 17$

```
E<> ((Person1.L_P == 1 and
      Person2.L_P == 1 and
      Person3.L_P == 1 and
      Person4.L_P == 1)
and global == 17)
```

É impossível que todos os aventureiros estejam no outro lado da ponte em menos de 17 minutos: $A \Box \neg((\forall_{1 \leq i \leq 4} Person_i.L_P = 1) \wedge global < 17)$

```
A[] not ((Person1.L_P == 1 and
  Person2.L_P == 1 and
  Person3.L_P == 1 and
  Person4.L_P == 1)
and global < 17)
```

4 Conclusão

Após a demonstração da nossa abordagem a este problema, dá-se por concluído este trabalho prático.

De um modo geral estamos satisfeitos com o porque fizemos tudo o que foi pedido, e foi uma forma de conseguirmos aplicar os conteúdos que fomos aprendendo e assimilando ao longo desta UC, tanto na primeira parte do trabalho (modelação), como na segunda (verificação).

Numa nota final, podemos concluir que de facto a modelação é uma ferramenta forte porque nos permite investigar, testar ou até mesmo prever o comportamento de sistemas de vida real, sem que tenhamos de gastar recursos a construí-lo.