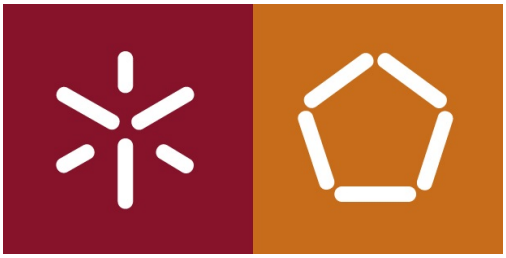

Universidade do Minho

Mestrado Integrado em Engenharia Informática

Arquitetura e Cálculo - 2019/20

Modelação e análise de um sistema de tempo real

Carla Cruz a80564

Jorge Cruz a78895

12 de Maio de 2020

1 Introdução

No âmbito da unidade curricular de *Arquitetura e Cálculo*, foi desenvolvido um modelo do sistema, bem como, verificação e teste com base no problema proposto. Para tal foi necessário aplicar o conhecimento adquirido relativamente à ferramenta **UPPAAL** e autómatos temporais.

Neste relatório iremos apresentar toda a linha de pensamento e estratégias utilizadas para a concretização dos objetivos, desde a descrição do problema apresentado no enunciado, passando pela apresentação da forma como implementamos a solução e, por fim, as conclusões obtidas.

2 Descrição do Problema

O contexto do problema consiste num grupo de quatro aventureiros que, durante a noite, encontraram uma ponte frágil sob um precipício profundo que necessitam de atravessar. Por razões de segurança, decidiram que não devem atravessar a ponte mais do que 2 pessoas ao mesmo tempo e que um deles deve transportar a lanterna para iluminar o caminho, sendo que apenas existe uma lanterna com eles.

Apresentando habilidades diferentes, os mesmos demoram tempos diferentes a percorrer a ponte, demorando **1, 2, 5 e 10 minutos**, respectivamente. Cada par de aventureiros que atravesse a ponte simultaneamente demora o período de tempo igual ao do mais lento dos dois aventureiros.

Os aventureiros têm opiniões diferentes relativamente ao tempo que demoram todos a atravessar esta ponte. Um afirma que eles não podem estar do outro lado em menos de 19 minutos. Outro discorda e afirma que isso pode ser feito em 17 minutos. Como nossa tarefa, iremos verificar através da ferramenta **UPPAAL**, os seguintes aspetos:

- Modelar o sistema com base na situação apresentada.
- Expressar em CTL as opiniões de cada aventureiro.
- Testar as fórmulas que provam a sua veracidade.

3 Modelo do Sistema

Para facilitar a modelação deste sistema foi utilizada a composição paralela, pelo que foi realizado um autómato para a lanterna e outro relativo a cada pessoa. Comecemos pelo autómato da lanterna.

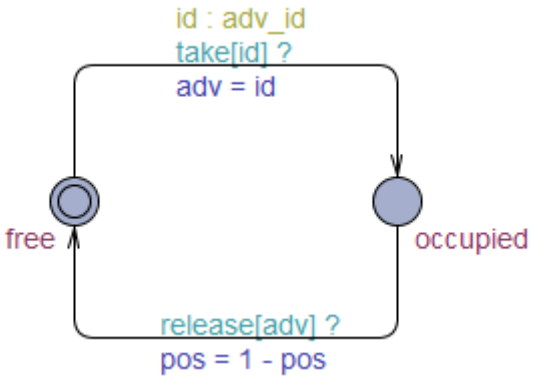

Figura 2: Autómato da lanterna

O comportamento da lanterna foi muito simples de modelar porque apresentava apenas dois estados diferentes e estava dependente das ações dos aventureiros. Ou seja, por outras palavras, a lanterna pode estar livre (estado ***free***) ou pode estar a ser usada por uma das pessoas (estado ***occupied***), pelo que a troca de estados acontece quando alguém pega ou larga a lanterna. Uma das escolhas acerca deste modelo é que, a partir do momento que um aventureiro pega na lanterna, este obrigatoriamente terá de atravessar a ponte

e libertar a lanterna no lado contrário. Isto é importante por dois motivos. Primeiro, facilitou o controlo da localização da lanterna, em que apenas é necessário trocar quando esta é largada por um dos aventureiros, ou seja, quando o seu estado passa de ocupada para livre. A posição consiste num inteiro que pode assumir os valores 0 e 1 para representar o início e o fim da ponte, respetivamente. Segundo, garantiu uma das restrições deste problema que consistia na exclusão mútua sobre a lanterna, ou seja, em que apenas um dos aventureiros pode segurar a lanterna de cada vez. A partir do momento em que um aventureiro pega na lanterna (ação ***take***), a próxima ação sobre a lanterna apenas poderá ser a de ***release*** e o aventureiro, ao atravessar a ponte, terá de obrigatoriamente exercer esta ação para chegar ao lado contrário. Notar que apenas o aventureiro que segura a lanterna pode libertá-la. Isto acontece porque existe um canal ***take*** e um canal ***release*** para cada um, ou seja, a lanterna ao ser ocupada pelo o canal ***take*** do aventureiro 1 apenas pode ser libertada pelo ***release*** do aventureiro 1. Para perceber melhor o sistema, observemos o autómato para cada um dos aventureiros.

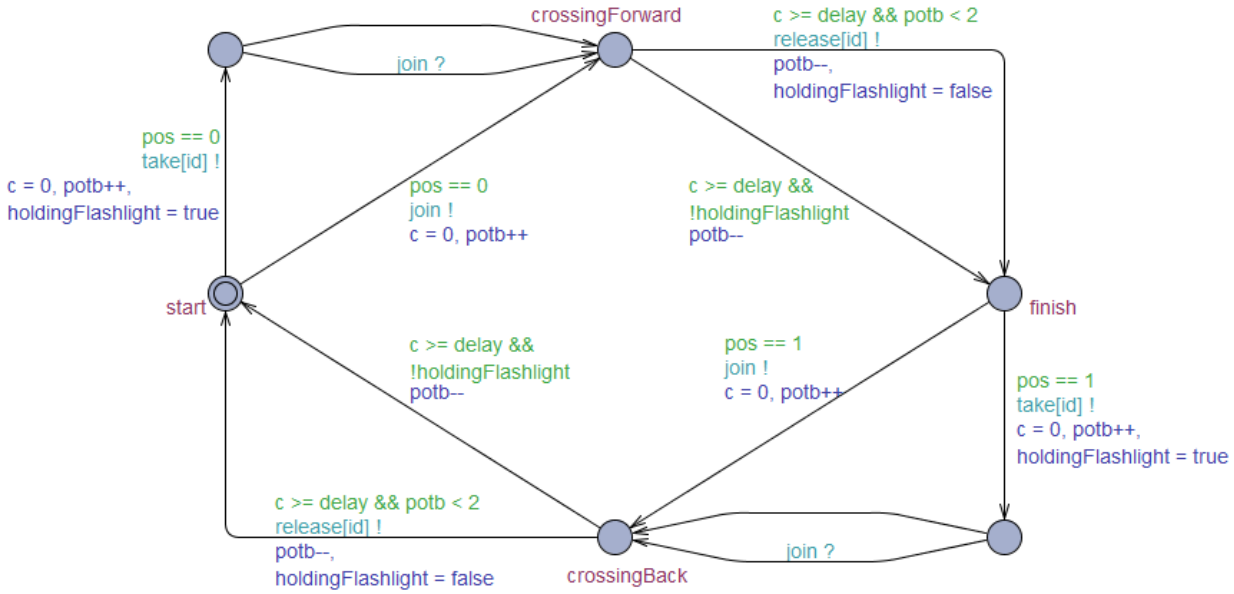

Figura 3: Autómato do aventureiro

Primeiramente, para modelar o comportamento de cada aventureiro foi necessário não assumir quaisquer restrições de tempo. Assim, os principais obstáculos a ultrapassar seriam impedir mais do que duas pessoas a atravessar a ponte de cada vez e que, obrigatoriamente, uma dessas pessoas transportasse a lanterna. Se olharmos para o estado inicial (estado ***start***), existem apenas duas ações que um aventureiro pode realizar: ele pode juntar-se a alguém para atravessar a ponte (ação ***join***) ou pode pegar na lanterna (caso esteja do seu lado) e avançar para um estado intermédio no qual decide se espera que alguém se junte a ele. Observando com melhor atenção, quando alguém decide atravessar a ponte, necessita de, ou obter a lanterna, ou existir um aventureiro com a lanterna que esteja à sua espera para se juntar através da sincronização do canal ***join***. Após dois aventureiros chegarem ao estado ***crossingForward*** (ou ***crossingBack***), é impossível haver um terceiro indivíduo a juntar-se já que os dois caminhos que levam a esse estado estão impedidos (o terceiro aventureiro não consegue obter a lanterna e, devido a isto, não consegue também sincronizar-se com alguém através do canal ***join*** pois não existe ninguém disponível). Ou seja, não só estamos a obrigar a uma das pessoas a segurar a lanterna, como estamos a limitar o número de aventureiros a atravessar a ponte.

Após atravessar a ponte, é importante referir que um dos aventureiros terá de libertar a lanterna antes de chegar ao lado contrário (devido à razão dada na explicação do autómato da lanterna). Porém, tal não pode acontecer antes do aventureiro que o acompanha passar também para o lado oposto pois, caso contrário, poderia acontecer que dois aventureiros estivessem a atravessar a ponte em sentidos opostos (ou seja, um aventureiro no estado ***crossingForward*** e um aventureiro no estado ***crossingBack***). Para impedir isto de acontecer, foi necessário a existência de uma variável com o número de pessoas a atravessar a ponte (***potb*** - *People On The Bridge*) em conjunto com a guarda $potb < 2$ na ação ***release***.

3.1 Expressões em CTL

Um processo importante durante a modelação e análise de um sistema de tempo real é a sua verificação. Neste sentido, foi necessário definir algumas expressões em CTL a que nosso modelo deve responder. Foi preciso definir as expressões baseadas nas teorias dos nossos aventureiros apresentadas no enunciado do projeto. Para além destas, foi necessário garantir outras que são importantes para o modelo definido em **UPPAAL**. Assim, iremos apresentar as propriedades definidas.

3.1.1 Propriedade 1

Uma das propriedades mais importantes a garantir de início era a ausência de *deadlocks* no nosso sistema. Assim, a primeira propriedade garante que, para todos os estados possíveis, existe sempre um caminho pelo qual o sistema pode evoluir.

$$A\Box \quad \neg(deadlock)$$

3.1.2 Propriedade 2

Uma propriedade que também consideramos importante no nosso modelo é a propriedade que garante que existe um caminho em que eventualmente todos os aventureiros chegam ao outro lado da ponte, ou seja, ao estado ***finish***. Por outras palavras, ser possível chegar a soluções para o problema.

$$E\Diamond \quad \forall_{i:int[1,4]}\ Adventurer(i).finish$$

3.1.3 Propriedade 3

Esta propriedade pretende verificar que é possível todos os aventureiros estarem do outro lado em 17 minutos, sendo que corresponde à opinião de um dos aventureiros. Desta forma, temos de garantir que existe um caminho em que eventualmente todos os aventureiros estarão no estado ***finish*** e que o tempo global passado será igual a 17 minutos.

$$E\Diamond \quad \forall_{i:int[1,4]}\ Adventurer(i).finish\ \wedge\ time == 17$$

3.1.4 Propriedade 4

Na quarta propriedade pretende-se verificar que é impossível todos os aventureiros estarem do outro lado em menos de 17 minutos. Desta forma, temos de garantir que não existe nenhum estado possível do sistema em que todos os aventureiros estão na localização ***finish*** e o tempo global passado é menor que 17 minutos.

$$A\Box \quad \neg(\forall_{i:int[1,4]}\ Adventurer(i).finish\ \wedge\ time < 17)$$

3.1.5 Propriedade 5

Um dos pontos importantes do enunciado era garantir a exclusão mútua sobre a lanterna. Neste sentido, com a próxima propriedade, mostramos que, em todos os estados possíveis do sistema, se existirem dois aventureiros a segurar a lanterna, esses aventureiros têm de ser o mesmo.

$$A\Box \quad \forall_{i:int[1,4]}\forall_{j:int[1,4]}\ Adventurer(i).holdingFlashlight \wedge Adventurer(j).holdingFlashlight \implies i == j$$

3.1.6 Propriedade 6

Uma das pistas para chegar à solução do problema seria obrigar as duas pessoas mais lentas a atravessar a ponte juntas. Assim, tentamos perceber se era impossível existir uma solução em que estas pessoas atravessam separadamente e o tempo final de todo o percurso ser 17 minutos. Definimos então uma propriedade em que, caso num dos estados do sistema os aventureiros mais lentos estejam em localizações diferentes (o que significa que atravessaram separadamente), então, nos estados posteriores a esse, nunca será verdade estarem todos os aventureiros na localização ***finish*** e o tempo global ser 17 minutos.

$$((Adventurer(3).start\ \wedge Adventurer(4).finish)\ \vee (Adventurer(3).finish\ \wedge Adventurer(4).start))$$
$$\rightsquigarrow\ \neg(\forall_{i:int[1,4]}\ Adventurer(i).finish \wedge time == 17)$$

3.1.7 Propriedade 7

Uma propriedade que faria todo definir e muito importante é a impossibilidade de uma pessoa atravessar em frente e outra no sentido contrário, ao mesmo tempo. Assim provamos que não existe nenhum estado possível do sistema em que isto aconteça, podendo apenas um dos percursos (*Forward* ou *Back*) estar a ser percorrido.

$$A\Box \quad \neg(\forall_{i:int[1,4]}\forall_{j:int[1,4]} \ (Adventurer(i).crossingForward \ \wedge Adventurer(j).crossingBack))$$

3.1.8 Propriedade 8

Um requisito deste problema é que a lanterna deve ser transportada por um dos aventureiros que atravessa a ponte. Assim, definimos uma propriedade que, caso um aventureiro esteja a passar a ponte sozinho, deve transportar a lanterna consigo e, caso este não a tenha consigo, significa que está acompanhado pelo o portador da lanterna.

- No sentido do *Start* para *Finish*

$$A\Box \quad (\forall_{i:int[1,4]} \ Adventurer(i).crossingForward \implies (Adventurer(i).holdingFlashlight \vee (\exists_{j:int[1,4]} \ Adventurer(j).crossingForward \wedge Adventurer(j).holdingFlashlight))$$

- No sentido do *Finish* para *Start*

$$A\Box \quad (\forall_{i:int[1,4]} \ Adventurer(i).crossingBack \implies (Adventurer(i).holdingFlashlight \vee (\exists_{j:int[1,4]} \ Adventurer(j).crossingBack \wedge Adventurer(j).holdingFlashlight))$$

3.2 Teste do Modelo

Após ter sido criado o modelo e todas as expressões e propriedades necessárias, estas tiveram de ser verificadas de modo a testar se o sistema responde de forma positiva, cumprindo estas propriedades. Na próxima imagem, através da ferramenta **UPPAAL**, podemos observar que todas as propriedades referidas são verdadeiras.

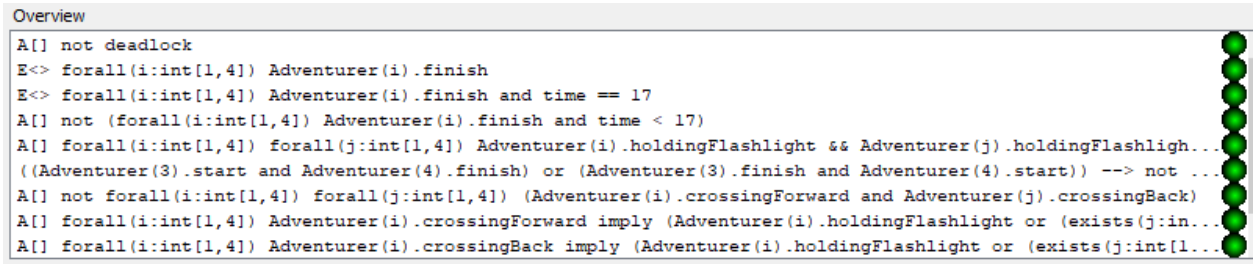

Figura 4: Verificação das propriedades na ferramenta **UPPAAL**

Para realizar o teste do modelo e após provar que é possível todos os aventureiros chegarem ao lado do *finish* da ponte em 17 minutos, conseguimos obter o *trace* que nos indica todo o percurso que os aventureiros fizeram para conseguir dar resposta a esta propriedade estabelecida.

Com base na animação que retratava o problema, previamente disponibilizada pelos professores, é possível verificar que o *trace* obtido corresponde ao apresentado no vídeo ilustrativo. Concluímos, com sucesso, que é possível validar a teoria do nosso aventureiro de chegar ao outro lado em 17 minutos:

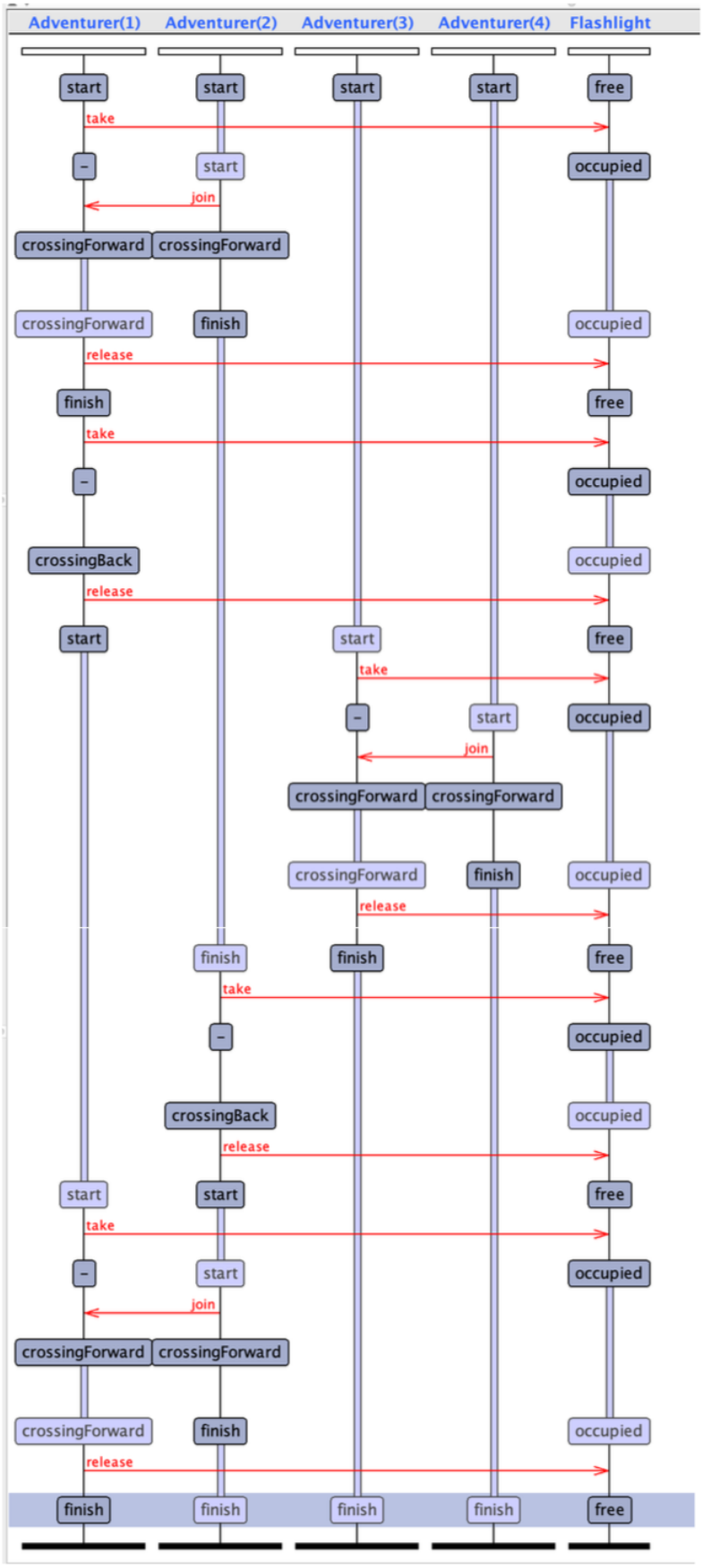

Figura 5: Percurso para chegarem em 17 minutos ao outro lado da ponte

4 Conclusão

Com a elaboração deste projeto, concluímos que utilizar a ferramenta **UPPAAL** pode ser muito útil na modelação e análise de sistemas de tempo real. Porém, o desenvolvimento do modelo do sistema pode revelar-se laborioso devido à sua complexidade e, por vezes, é necessário tomar decisões quanto ao modelo final. Durante a elaboração deste trabalho, vários modelos foram construídos e as suas diferenças encontravam-se, essencialmente, na quantidade de estados e variáveis necessárias para uma correta representação do sistema. No fim, as nossas escolhas quanto ao modelo final foram aquelas que permitiram e facilitaram expressar as propriedades que pretendíamos testar no nosso sistema.

Uma das características do modelo final que nos orgulhamos de referir consiste na sua adaptação para um sistema mais complexo. Ou seja, é possível, por exemplo, acrescentar mais aventureiros ao modelo com grande facilidade e sem grandes modificações. Apesar de neste sistema não ser uma característica fundamental, noutros problemas é essencial o modelo ser o mais adaptável possível a diferentes propriedades.

Achamos importante referir que há sempre espaço para melhorar, desde tornar, por vezes, o modelo mais intuitivo a otimizá-lo com certas modificações cirúrgicas. Um exemplo claro disto, é impedir que um aventureiro sozinho vá do ***start*** para o ***finish*** dado que, esta ação não tem um efeito determinante para o sistema, podendo até trazer um efeito negativo pois aumenta o tempo total de todos os aventureiros chegarem ao outro lado.

Contudo, foi com o auxílio da ferramenta **UPPAAL** que conseguimos, com sucesso, dar resposta aos problemas do enunciado, reconhecendo que esta etapa foi concluída com sucesso e que tivemos a possibilidade de aplicar na prática todos os conhecimentos adquiridos nesta área de estudo.