

UFSC

DAS5203 - MODELAGEM E CONTROLE DE
SISTEMAS A EVENTOS DISCRETOS

Laboratório 4

Autores:

Ígor YAMAMOTO

Iago SILVESTRE

7 de Maio de 2016

Sumário

1	Roteiro	2
1.1	Descrição do Problema	2
1.2	Ferramenta	2
1.3	Reconhecimentos	2
2	Resolução	3

1 Roteiro

1.1 Descrição do Problema

O comportamento de John e Fred para ir à seus respectivos trabalhos é descrito a seguir:

John vai todo dia ao seu trabalho, ou de carro (a duração do trajeto sendo de 30 à 40 minutos) ou de onibus (a duração sendo no mínimo de 60 minutos). Fred vai trabalhar, ou de carro (duração do trajeto entre 20 e 30 minutos) ou de carona (de 40 à 50 minutos).

O cenário para o dia de hoje em termos de restrições temporais é o seguinte: Hoje, John saiu de casa entre 7h10 e 7h20 e Fred chegou ao local de trabalho entre 8h00 e 8h10. Além do mais, John chegou ao local de trabalho entre 10 e 20 minutos após Fred ter saído de sua casa.

- Modelar separadamente os comportamentos de John e de Fred.
- Modelar o relógio e o cronômetro que permitem representar as restrições temporais anteriores.
- Verificar que as restrições temporais do cenário anterior são coerentes com o comportamento de John e Fred, utilizando a lógica temporal para expressar a propriedade a verificar e a ferramenta SELT para verifica-la;
- Verificar ainda se, foi possível John pegar o onibus e Fred utilizar carona para este cenário, utilizando a lógica temporal e a ferramenta SELT; ·
- Realizar as duas verificações anteriores, utilizando um observador e a análise de alcançabilidade.

1.2 Ferramenta

A ferramenta a ser utilizada é a ferramenta TINA (<http://www.laas.fr/tina>) desenvolvida no LAAS-CNRS (Toulouse - France).

1.3 Reconhecimentos

Esta aula prática é inspirada pela aula prática intitulada "TP Modèles Temporels" do INSAToulouse, de autoria de François Vernadat (LAAS) .

2 Resolução

- Modelar separadamente os comportamentos de John e de Fred.
- Modelar o relógio e o cronômetro que permitem representar as restrições temporais anteriores.

Os modelos obtidos foram os seguintes(figura 1):

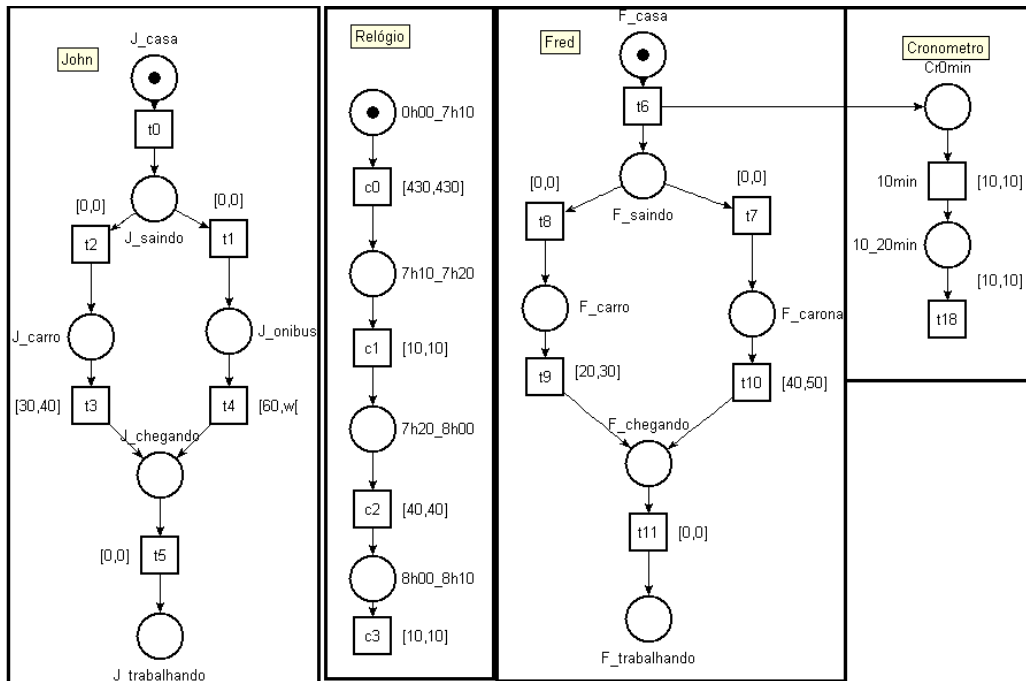


Figura 1: Modelos John,Relógio,Fred,Cronômetro

- Verificar que as restrições temporais do cenário anterior são coerentes com o comportamento de John e Fred, utilizando a lógica temporal para expressar a propriedade a verificar e a ferramenta SELT para verificá-la;

Relembrando, o cenário para o dia de hoje em termos de restrições temporais é o seguinte: Hoje, John saiu de casa entre 7h10 e 7h20 e Fred chegou ao local de trabalho entre 8h00 e 8h10. Além do mais, John chegou ao local de

trabalho entre 10 e 20 minutos após Fred ter saído de sua casa. Para testar esse cenário, o seguinte comando em LTL foi utilizado:

$$[] \neg ((J_saindo \wedge 7h10_7h20) \wedge \langle \rangle ((J_chegando \wedge 10_20min) \wedge (F_chegando \wedge 8h00_8h10)));$$

Esse comando retorna False e o seguinte contra exemplo:

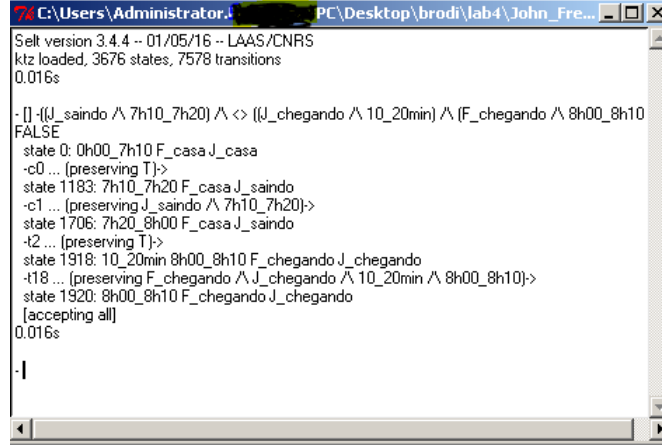


Figura 2: Contra exemplo cenário exemplo

Demonstrando que existe um caminho em que a propriedade é satisfeita.

- Verificar ainda se, foi possível John pegar o onibus e Fred utilizar carona para este cenário, utilizando a lógica temporal e a ferramenta SELT;

Para testar esse cenário, o seguinte comando foi utilizado:

$$[] \neg ((J_saindo \wedge 7h10_7h20) \wedge \langle \rangle ((J_chegando \wedge 10_20min) \wedge (F_chegando \wedge 8h00_8h10)) \wedge \langle \rangle J_onibus \wedge \langle \rangle F_carona);$$

Que retorna a resposta exibida na figura 3:

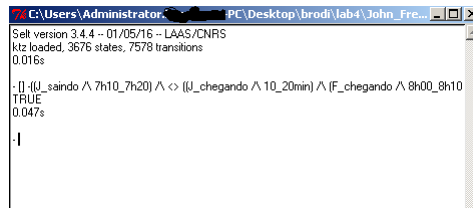


Figura 3: Resposta propriedade 2

Demonstrando que não há nenhum caminho para que john pega onibus e fred utiliza carona e o cenário descrito na propriedade 1 acontece.

- Realizar as duas verificações anteriores, utilizando um observador e a análise de alcançabilidade. Para esse item, mudamos a rede inicial com implementação dos observadores.

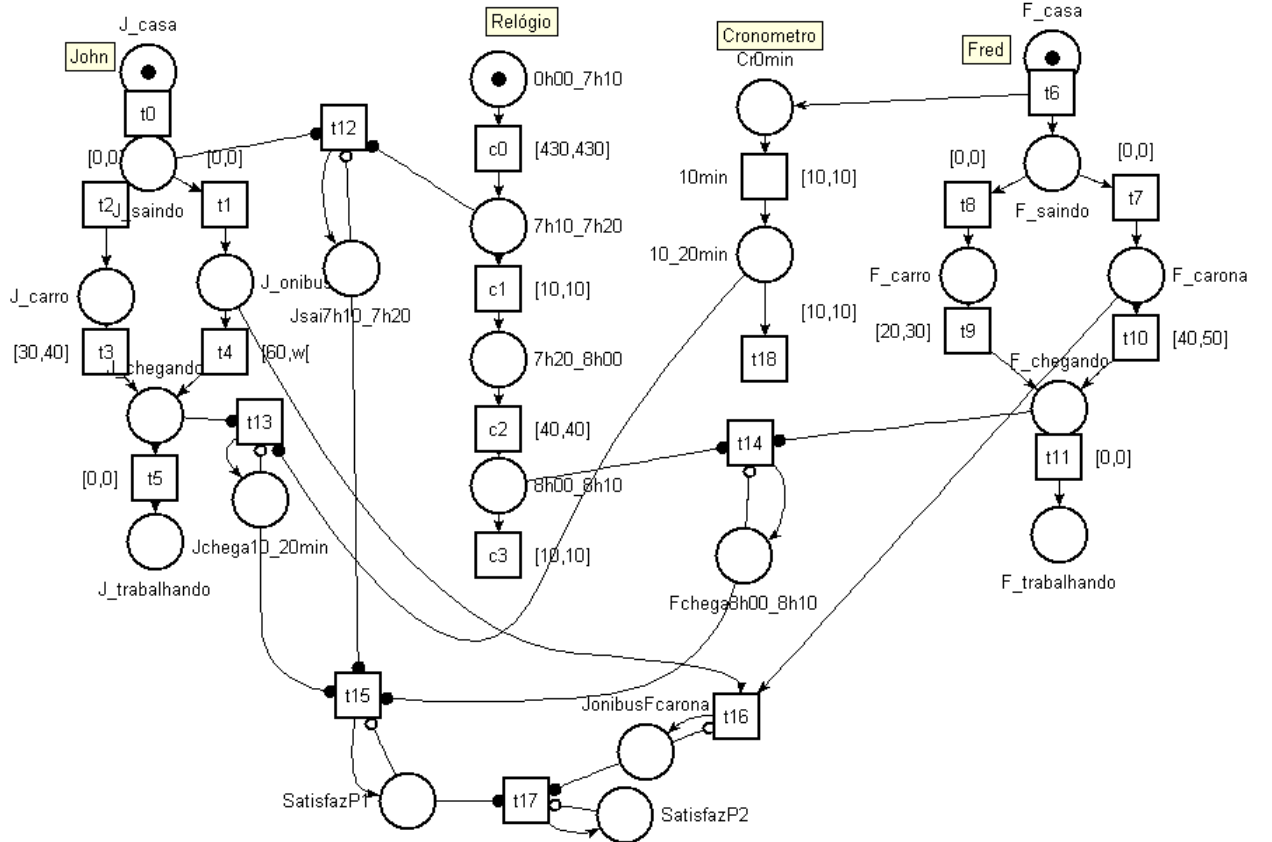


Figura 4: Rede com observadores

Os observadores adicionados são os seguintes:

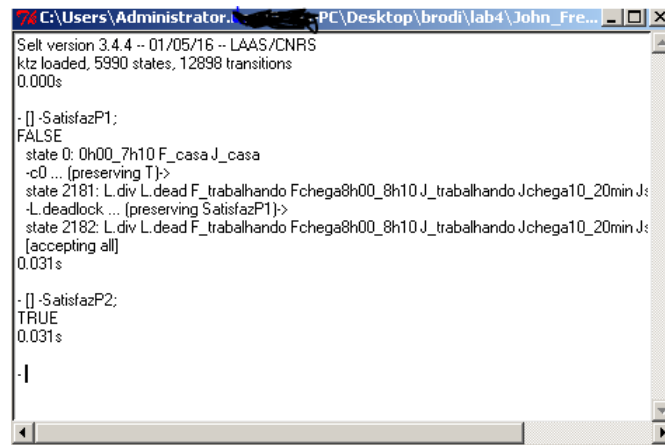
1. Jsai7h10_7h20 serve para checar se John saiu entre 7:10h e 7:20h.
2. Jchega10_20min serve para checar se John chegou ao trabalho de 10 à 20 minutos após Fred ter saído de casa.
3. Fchega8h00_8h10 serve para checar se Fred chegou no trabalho entre 8:00h e 8:10h

4. JonibusFcarona serve para checar a propriedade em que John utiliza o ônibus e Fred utiliza a carona.
5. SatisfazP1 : Quer dizer que o sistema satisfaz a primeira propriedade listada no problema
6. SatisfazP2 : Idem ao item anterior, porém para a segunda propriedade

Para verificar se o sistema satisfaz a primeira e segunda prioriedade, utilizaremos os seguintes códigos:

`[] -SatisfazP1;` e `[] -SatisfazP2;`

Como era esperado, o resultado obtido com os observadores foi identico ao resultado prévio.



```

C:\Users\Administrator\... PC\Desktop\brodi\lab4\John_Fre...
Self version 3.4.4 -- 01/05/16 -- LAAS/CNRS
ktz loaded, 5990 states, 12898 transitions
0.000s

- [] -SatisfazP1;
FALSE
state 0: 0h00_7h10 F_casa J_casa
-c0 ... (preserving T1)->
state 2181: L.div L.dead F_trabalhando Fchega8h00_8h10 J_trabalhando Jchega10_20min J:
-L.deadlock ... (preserving SatisfazP1)->
state 2182: L.div L.dead F_trabalhando Fchega8h00_8h10 J_trabalhando Jchega10_20min J:
[accepting all]
0.031s

- [] -SatisfazP2;
TRUE
0.031s

-I

```

Figura 5: Rede com observadores