Exame de *Formal Verification of Critical Applications* 2021-2022

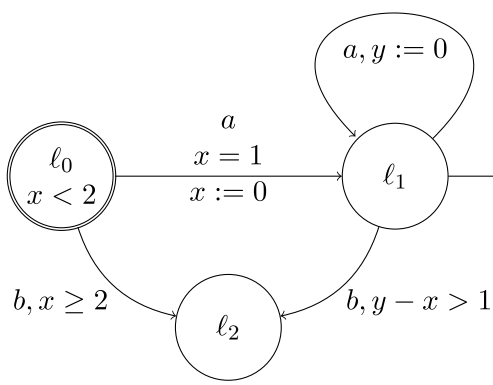
Mestrado em Engenharia de Sistemas Computacionais Críticos

Eduardo Tovar & David Pereira & José Proença

8 Julho 2022 (época normal) – duração: 2h

Modelação de sistemas de tempo real

Exercício 1. Considere o autómato de tempo real abaixo.

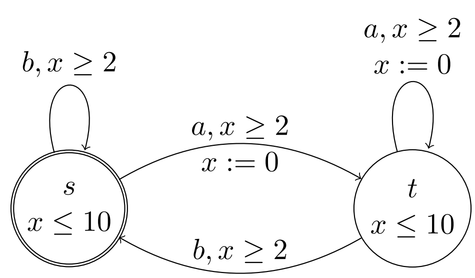


1.1. Defina o autómato formalmente como um tuplo (L, L_0, Act, Tr, Inv).

1.2. O autómato tem algum caminho (*trace*) com comportamento Zeno? Explique.

1.3. O autómato tem algum caminho (*trace*) com um *timelock*? Explique.

Exercício 2. Considere o autómato de tempo real abaixo.



2.1. Desenhe um autómato de tempo real com uma única localização que seja *timed bisimilar* ao autómato apresentado.

2.2. Apresente a bissimulação que mostra que os autómatos são bissimilares.

Exercício 3. Considere um sistema com 2 autómatos de tempo real em paralelo, *Semáforo* e *Botão*, com estados $\{Verde, Amarelo, Vermelho\}$ e $\{Carregado, Solto\}$, respetivamente. Assuma ainda que:

- o *Semáforo* tem um relógio v que é colocado a zero de cada vez que este chega ao estado $Verde$;
- o *Botão* tem um relógio c que é colocado a zero quando este é $Carregado$; e
- o *Botão*, depois de $Carregado$, fica $Solto$ quando o *Semáforo* ficar $Verde$.

Sem modelar os autómatos, formalize as seguintes propriedades usando lógica temporal (como em UPPAAL).

3.1. O *Botão* não pode estar $Carregado$ enquanto a luz está $Verde$.

3.2. A *Semáforo* pode ficar $Amarelo$.

3.3. Se o *Botão* for $Carregado$, demora no máximo 125 unidades de tempo até a luz ficar $Verde$.

Lógica e verificação dedutiva

Exercício 4. Considere os triplos de Hoare apresentados abaixo e, para cada um deles, calcule a respetiva pré-condição mais fraca usando o algoritmo introduzido nas aulas. Mostre que a pré-condição mais fraca é satisfeita pela pré-condição explicitada no triplo.

1. $\{true\}$ if(x > 0) then {y := x;} else {y := -x;} $\{y \geq 0\}$
2. $\{x \neq y\}$ t := x; x := y; y := t $\{x \neq y\}$

Exercício 5. Considerando os mesmo triplos de Hoare apresentados na questão anterior, aplique um dos algoritmos de geração de obrigações de demonstração/prova (introduzidos nas aulas como VC e VCG) a cada um desses triplos. **Nota:** No anexo C podem encontrar um dos algoritmos introduzidos na disciplina. A utilização desse algoritmo para efeitos da resolução desta questão será sujeita a uma penalização de 25%; a utilização do algoritmo alternativo não será sujeita a qualquer penalização.

Exercício 6. Considere o triplo de Hoare apresentado abaixo.

```
{n = n0 ∧ n0 >= 0}
x := 0;
while(n != 0) {
 x := x + 1;
 n := n - 1;
}
{x = n0}
```

Da lista de opções apresentada abaixo, indique aquela que é uma invariante válida e que garante a correção do triplo. Justifique.

1. $n \geq 0 \wedge x \geq 0$
2. $x + n0 = n$
3. $x + n = n0$

A Anexo: Semântica de Timed Automata

Let $ta = \langle L, L_0, Act, C, Tr, Inv \rangle$

$$\mathcal{T}(ta) \;=\; \langle S, S_0 \subseteq S, N, T \rangle$$

where

- $S = \{\langle l, \eta \rangle \in L \times (\mathbb{R}_0^+)^C \mid \eta \models Inv(l)\}$
- $S_0 = \{\langle \ell_0, \eta \rangle \mid \ell_0 \in L_0 \; e \; \eta \, x = 0$ for all $x \in C\}$
- $N = Act \cup \mathbb{R}_0^+$ (ie, transitions can be labelled by actions or delays)
- $T \subseteq S \times N \times S$ is given by:

$$\begin{array}{rcl} \langle l, \eta \rangle \xrightarrow{a} \langle l', \eta' \rangle & \Leftarrow & \exists_{l \xrightarrow{g,a,U} l' \in Tr} \; \eta \models g \;\wedge\; \eta' = \eta[U] \;\wedge\; \eta' \models Inv(l') \\ \langle l, \eta \rangle \xrightarrow{d} \langle l, \eta + d \rangle & \Leftarrow & \exists_{d \in \mathbb{R}_0^+} \; \eta + d \models Inv(l) \end{array}$$

A *path* of a timed automata ta is a (possibly empty) trace of $\mathcal{T}(ta)$: $\langle \ell_1, \eta_1 \rangle \xrightarrow{\alpha_1} \langle \ell_2, \eta_2 \rangle \xrightarrow{\cdots} \langle \ell_n, \eta_n \rangle$.

B Anexo: Timed bisimulation

A relation R is an timed simulation iff whenever $s_1 R s_2$, for any action a and delay $d \in \mathbb{R}_0^+$,

$$\begin{array}{l} s_1 \xrightarrow{a} s_1' \;\Rightarrow\; \text{there is a transition} \quad s_2 \xrightarrow{a} s_2' \;\&\; s_1' R s_2' \\ s_1 \xrightarrow{d} s_1' \;\Rightarrow\; \text{there is a transition} \quad s_2 \xrightarrow{d} s_2' \;\&\; s_1' R s_2' \end{array}$$

And it is an timed bisimulation if its converse is also an untimed simulation.

C Anexo: Geração de condições de demonstração/prova

$$\begin{array}{rcl} VC(\{P\}\, skip \,\{Q\}) & = & \{P \rightarrow Q\} \\ VC(\{P\}\, x := E \,\{Q\}) & = & \{P \rightarrow Q[E \mapsto x]\} \\ VC(\{P\} C_1; C_2 \{Q\}) & = & VC(\{P\}\, C_1 \,\{wprec(C_2; Q)\}) \\ & & \cup \\ & & VC(\{wprec(C_2, Q)\}\, C_2 \,\{Q\}) \\ VC(\{P\}\, if(B)\, then\, C_1\, else\, C_2 \,\{Q\}) & = & VC(\{P \wedge B\}\, C_1 \,\{Q\}) \\ & & \cup \\ & & VC(\{P \wedge \neg B\}\, C_2 \,\{Q\}) \\ VC(\{P\}\, while(B)\, \{I\}\, C\{\}) & = & \{P \rightarrow I, I \wedge \neg B \rightarrow Q\} \\ & & \cup \\ & & VC(\{P \wedge \neg B\}\, C \,\{Q\}) \end{array}$$