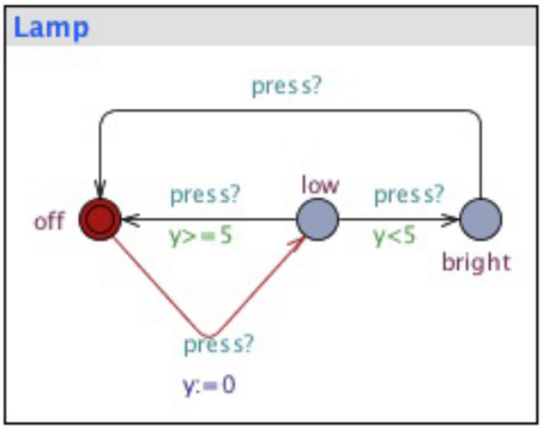
Arquitetura e Cálculo - TPC 2

Luís Ribeiro (A85954)
Mestrado em Engenharia Informática
Universidade do Minho

1 Exercício 1

Define $< L, L_0, Act, C, Tr, Inv >$.



$L = \{off, low, bright\}$
$L_0 = \{off\}$

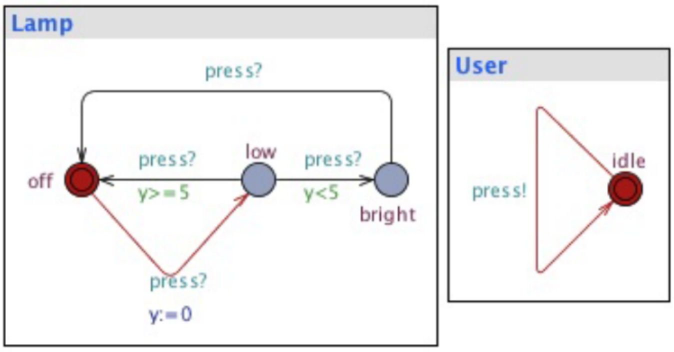
$Act = \{press?\}$
$C = \{y\}$

$$
\begin{aligned}
Tr = \{&(off, true, press?, \{y\}, low),\\
&(low, y >= 5, press?, \{\}, off),\\
&(low, y < 5, press?, \{\}, bright),\\
&(bright, true, press?, \{\}, off)\}
\end{aligned}
$$

$Inv = \{x \rightarrow true \mid x \in L\}$

2 Exercício 2

Define the T_a of the composition.



$L_1 \times L_2 = \{(off, idle), (low, idle), (bright, idle)\}$
$L_{0,1} \times L_{0,2} = \{(off, idle)\}$

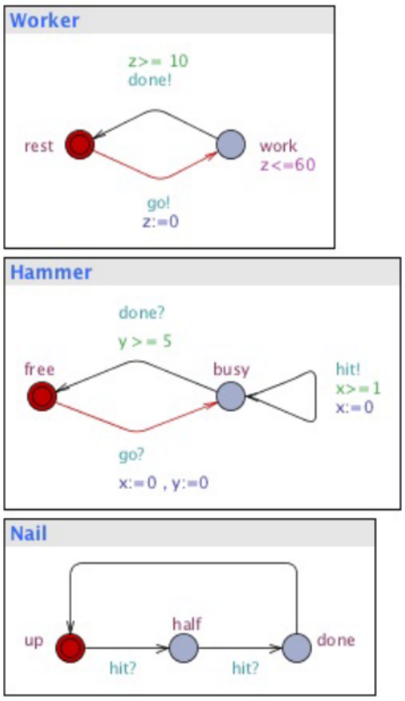
$Act_{\|_H} = \{\tau\_press\}$
$C_1 \cup C_2 = \{y\}$

$$
\begin{aligned}
Tr_{\|_H} = \{&((off, idle), true, \tau\_press, \{y\}, (low, idle)),\\
&((low, idle), y >= 5, \tau\_press, \{\}, (off, idle)),\\
&((low, idle), y < 5, \tau\_press, \{\}, (bright, idle)),\\
&((bright, idle), true, \tau\_press, \{\}, (off, idle))\}
\end{aligned}
$$

$Inv_{\|_H} = \{(off, idle) \rightarrow true, (low, idle) \rightarrow true, (bright, idle) \rightarrow true\}$

3 | Exercício 3

Define the T_a of the composition.



Para facilitar este processo, primeiro defini a composição paralela entre o *Worker* e *Hammer*. Em seguida, fiz a composição do T_a anterior, com o *timed automata Nail*.
Como existe uma transição não etiquetada, sendo esta em *Nail*, de *done* $\rightarrow$ *up*, dei-lhe o nome de `random`.
Isto terá sido em conta aquando a composição paralela $\|_G$ com *Nail*.
Temos assim, $(Worker \; \|_H \; Hammer) \; \|_G \; Nail$.

3.1 $Worker \; \|_H \; Hammer$

$L_{Worker} \times L_{Hammer} = \{(rest, free), (rest, busy), (work, free), (work, busy)\}$
$L_{0,Worker} \times L_{0,Hammer} = \{(rest, free)\}$

$Act_{\|_H} = \{\tau\_done, \tau\_go, hit!\}$
$C_{Worker} \cup C_{Hammer} = \{x, y, z\}$

$$
\begin{aligned}
Tr_{\|_H} = \{&((rest, free), true, \tau_{go}, \{x, y, z\}, (work, busy)),\\
&((work, busy), z \geq 10 \wedge y \geq 5, \tau_{done}, \{\}, (rest, free)),\\
&((work, busy), x \geq 1, hit, \{x\}, (work, busy)),\\
&((rest, busy), x \geq 1, hit, \{x\}, (rest, busy))\}
\end{aligned}
$$

$$
Inv_{\|_H} = \{(rest, free) \to true, (rest, busy) \to true, (work, free) \to z \leq 60, (work, busy) \to z \leq 60\}
$$

3.2 $(Worker \parallel_H Hammer) \parallel_G Nail$

$$
\begin{aligned}
L_H \times L_{Nail} = \{&(rest, free, up), (rest, free, half), (rest, free, done),\\
&(rest, busy, up), (rest, busy, half), (rest, busy, done),\\
&(work, free, up), (work, free, half), (work, free, done),\\
&(work, busy, up), (work, busy, half), (work, busy, done)\}
\end{aligned}
$$

$$
L_{0,H} \times L_{0,Nail} = \{(rest, free, up)\}
$$

$$
C_H \cup C_{Nail} = \{x, y, z\}
$$

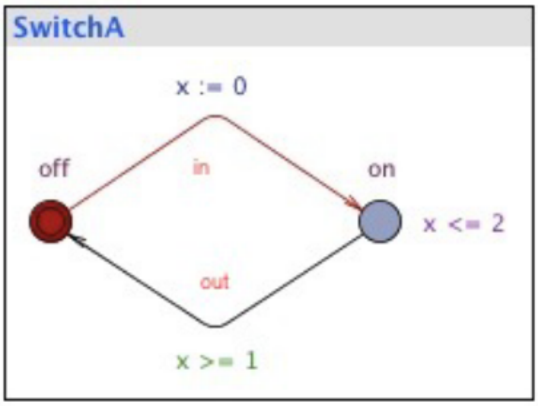
$$
Act_{\|G} = \{\tau_{go}, \tau_{done}, \tau_{hit}, random\}
$$

$$
\begin{aligned}
Tr_{\|_G} = \{&\\
&((rest, busy, up), x \geq 1, \tau\_hit, \{x\}, (rest, busy, half)),\\
&((rest, busy, half), x \geq 1, \tau\_hit, \{x\}, (rest, busy, done)),\\
&((work, busy, up), x \geq 1, \tau\_hit, \{x\}, (work, busy, half)),\\
&((work, busy, half), x \geq 1, \tau\_hit, \{x\}, (work, busy, done)),\\
\\
&((rest, free, up), true, \tau_{go}, \{x, y, z\}, (work, busy, up)),\\
&((rest, free, half), true, \tau_{go}, \{x, y, z\}, (work, busy, half)),\\
&((rest, free, done), true, \tau_{go}, \{x, y, z\}, (work, busy, done)),\\
\\
&((work, busy, up), y \geq 5 \wedge z \geq 10, \tau_{done}, \{\}, (rest, free, up)),\\
&((work, busy, half), y \geq 5 \wedge z \geq 10, \tau_{done}, \{\}, (rest, free, half)),\\
&((work, busy, done), y \geq 5 \wedge z \geq 10, \tau_{done}, \{\}, (rest, free, done)),\\
\\
&((rest, free, done), true, random, \{\}, (rest, free, up)),\\
&((rest, busy, done), true, random, \{\}, (rest, busy, up)),\\
&((work, free, done), true, random, \{\}, (work, free, up)),\\
&((work, busy, done), true, random, \{\}, (work, busy, up))\\
\}&
\end{aligned}
$$

$$
\begin{aligned}
Inv_{\|_G} = &\{(work, h, n) \to z \leq 60 \mid h \in L_{Hammer} \wedge n \in L_{Nail}\}\\
&\cup\\
&\{(w, h, n) \to true \mid w \in L_{Worker} \setminus \{work\} \wedge h \in L_{Hammer} \wedge n \in L_{Nail}\}
\end{aligned}
$$

4 Exercício 4

Define $\mathcal{T}(SwitchA)$.



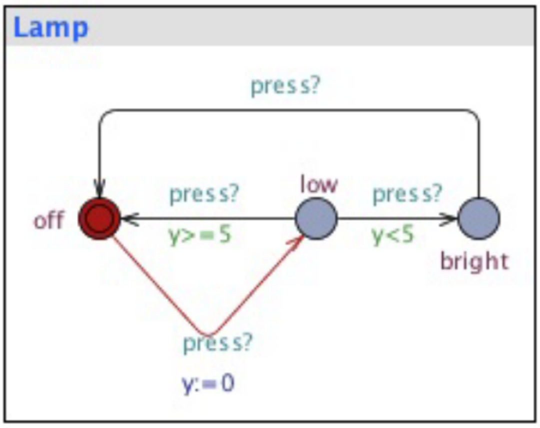
$S = \{\langle off, t\rangle \mid t \in \mathbb{R}_0^+\} \cup \{\langle on, t\rangle \mid 0 \leq t \leq 2\}$
$S_0 = \{(off, const0)\}$

$N = \{in, out\} \cup \mathbb{R}_0^+$

$$T = \{\forall_{t,d\geq 0} : (\langle off, t\rangle, d, \langle off, t+d\rangle)\}$$
$$\cup$$
$$\{\forall_{t\geq 0} : (\langle off, t\rangle, in, \langle on, 0\rangle)\}$$
$$\cup$$
$$\{\forall_{t,d\geq 0 \wedge t+d\leq 2} : (\langle on, t\rangle, d, \langle on, t+d\rangle)\}$$
$$\cup$$
$$\{\forall_{1\leq t\leq 2} : (\langle on, t\rangle, out, \langle off, t\rangle)\}$$

5 Exercício 5

Write 3 possible traces with different nr. of actions.



$\{\}$: Representa o Traço Vazio.

$\{\langle 0, press\rangle, \langle 6, press\rangle, \langle 7, press\rangle, \langle 10, press\rangle, \langle 11, press\rangle\}$

Este traço representa o seguinte caminho:

$$\langle off, 0\rangle \xrightarrow{press} \langle low, 0\rangle \xrightarrow{d=6} \langle low, 6\rangle \xrightarrow{press} \langle off, 6\rangle \xrightarrow{d=1} \langle off, 7\rangle \xrightarrow{press} \langle low, 0\rangle \xrightarrow{d=3} \langle low, 3\rangle \xrightarrow{press} \langle bright, 3\rangle \xrightarrow{d=1} \langle bright, 4\rangle \xrightarrow{press} \langle off, 4\rangle$$

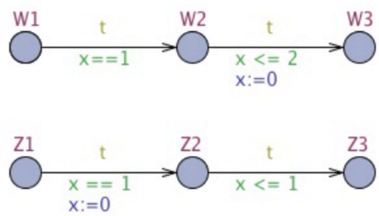
$\{\langle 1, press\rangle, \langle 3, press\rangle\}$

Este traço representa o seguinte caminho:

$$\langle off, 0\rangle \xrightarrow{d=1} \langle off, 1\rangle \xrightarrow{press} \langle low, 0\rangle \xrightarrow{d=3} \langle low, 3\rangle \xrightarrow{press} \langle bright, 3\rangle$$

6 Exercício 6

Are these timed-language equivalent? Explain.



Primeiro, vamos indicar os traços possíveis do LTS W e de Z separadamente, de forma a verificar se são equivalentes.

Traços de W:

- $\{\}$: Traço Vazio.
- $\{\langle 1, t\rangle\}$: $W_1 \xrightarrow[x==1]{t} W_2$. Como W_1 transita por t para W_2, só quando $x = 1$, o único traço possível é este.
- $\{\forall_{1 \leq d \leq 2} : \langle 1, t\rangle, \langle d, t\rangle\}$: $W_1 \xrightarrow[x==1]{t} W_2 \xrightarrow[x \leq 2; x:=0]{t} W_3$. W_2 transita para W_3, se $x \leq 2$. Como o x funciona como um *time-stamp*, que é sempre incrementado e não é reposto a 0 na transição W_1 para W_2, as restrições de d são as de x.

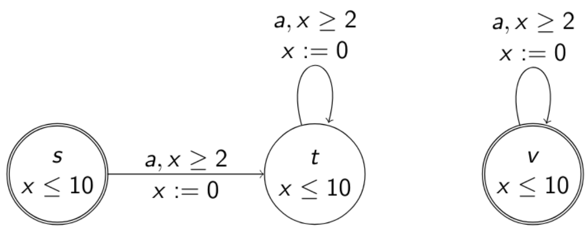
Traços de Z:

- $\{\}$: Traço Vazio.
- $\{\langle 1, t\rangle\}$: $Z_1 \xrightarrow[x==1; x:=0]{t} Z_2$. Como Z_1 transita por t para Z_2, só quando $x = 1$, o único traço possível é este. Nesta transição o *clock* x é reposto a 0 no estado W_2.
- $\{\forall_{1 \leq d \leq 2} : \langle 1, t\rangle, \langle d, t\rangle\}$: $Z_1 \xrightarrow[x==1; x:=0]{t} Z_2 \xrightarrow[x \leq 1]{t} Z_3$. W_2 transita para W_3, se $x \leq 2$. Neste caso, o x não funciona como um $time - stamp$, visto que é reposto a 0 na transição descrita acima. No entanto, a restrição de transição de Z_2 para Z_3 é de $x \leq 1$, significando que, apesar de x em Z_2 ser 0, só poderá ser incrementado até uma unidade (tal como em W) para permitir a transição para Z_3, portanto os limites do *time-stamp* d serão iguais aos impostos em W.

Tendo isto, podemos concluir que **são *timed-language equivalent***.

7 Exercício 7

Show a timed bisimulation with $\langle\langle s, \{x \mapsto 0\}\rangle, \langle v, \{x \mapsto 0\}\rangle\rangle$. If it exists, or explain why these states are not timed bisimilar.



$$
\begin{aligned}
R = \; & \{\langle\langle s, \{x \mapsto d\}\rangle, \langle v, \{x \mapsto d\}\rangle\rangle \mid 0 \leq d \leq 10\} \\
& \cup \\
& \{\langle\langle t, \{x \mapsto d\}\rangle, \langle v, \{x \mapsto d\}\rangle\rangle \mid 0 \leq d \leq 10\}
\end{aligned}
$$

De notar que s e v são bissimilares, porque tem a mesma transição por a com as mesmas restrições (*guards*), onde o *clock* x é reposto a 0. Tem também o mesmo invariante de estado, onde $x \leq 10$. Assim, o *time-stamp* d nestes estados, terá que respeitar a restrição imposta pelo invariante.

A transição de s por a para t, corresponde à transição por a de v para ele mesmo. O estado t respeita o mesmo invariante imposto por v, tendo também uma transição para ele mesmo por a com as mesmas *guards* e com o *reset* do *clock* x. Daí, t e v serem bissimilares entre eles.

Podemos concluir então que R se trata de uma ***Timed Bisimulation*** entre estes estados.