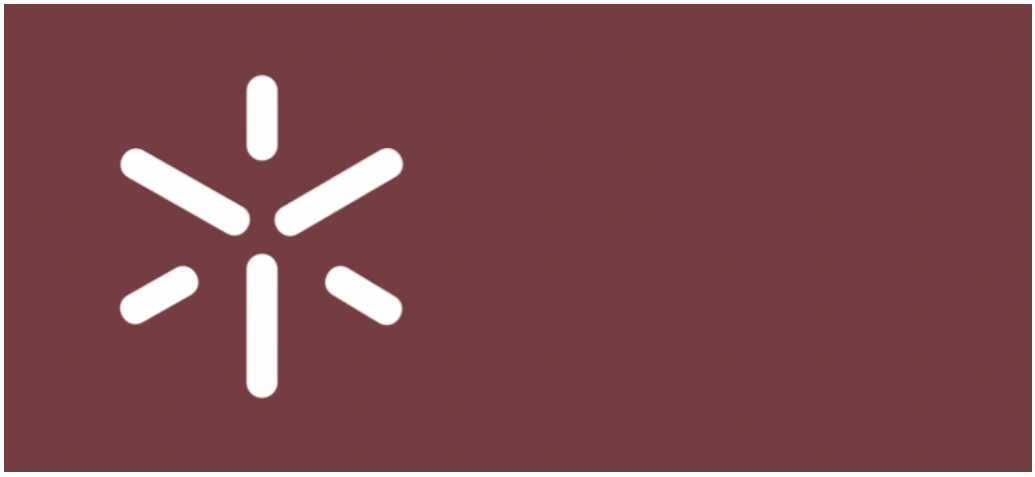

UNIVERSIDADE DO MINHO

Arquitetura e Cálculo

Exercícios Timed-Automata

Rafael Lourenço
(A86266)

Conteúdo

Capítulo 1

Resolução dos Exercícios

1.1 Exercício 1

```
R:<L,L0,Act,C,TR,inv>
L=off,low,bright
L0=off
Act = {press}
C   = {y}
Tr  = {t1,t2,t3,t4}
        t1=(off,true,press,{y},low)
        t2=(low,y>=5, press,{},off)
        t3=(low,y<5, press,{},bright)
        t4=(bright,true, press,{},off)
inv = {x->true | x in L}
```

Figura 1.1: Resolução

1.2 Exercício 2

```
R : Lamp || User = {L1 >< L2 , L0,1 >< L0,2,Act||h ,C1 U C2 ,Tr||h,Inv||h }
L1 >< L2 = {(off,idle) ,(low,idle) ,(bright,idle)}
L0,1 >< L0,2 = (off,idle)
Act||h= {tau_press}
C1 U C2 ={y}
Tr||h={t1,t2,t3,t4}
        t1=((off,idle)    --- ((true,true)), Tau_press,{y}) ---> (low,idle))
        t2=((low,idle)    --- ( (y>=5,true), Tau_press, {}) ---> (off,idle))
        t3=((low,idle)    --- ( (y<5, true), Tau_press, {}) ---> (bright,idle))
        t4=((bright,idle) --- ( (true,true), Tau_press, {}) ---> (off,idle))

Inv||h= {x->true | x in L1 >< L2}
```

Figura 1.2: Resolução

1.3 Exercício 3

```
Auxiliar:
(Worker || Hammer) = {L1 >< L2 , L0,1 >< L0,2,Act||h ,C1 U C2 ,Tr||h,Inv||h }
L1 >< L2 = {(rest,free) ,(work,busy)}
L0,1 >< L0,2 = (rest,free)
Act||h={Tau_go,Tau_done,hit}
C1 U C2={x,y,z}

Tr||h ={t1,t2,t3}
        t1=((rest,free)---  (true,Tau_go,{x,y,z})           ----->  (work,busy))
        t2=((work,busy)---  (x>=1,hit,{x})                          ----->  (work,busy))
        t3=((work,busy)---  ((z>=10,y>=5 ),done,{})  ----->  (rest,free))

((work, busy)---  ((y>=5, z>=10),(done!,done?),{y,z} ---> (rest, free))

((work,busy)---  (x>=1,hit,{x})  ----->  (Work,busy))

Inv||h =
A cor de rosa são os invariante.


R: (Worker || Hammer|| Nail) = {L1 >< L2 >< L3 , L0,1 >< L0,2 >< L0,3 ,Act||h||n ,C1 U
C2 U C3 ,Tr||h||n ,Inv||h||n }

L1 >< L2 >< L3 = {(rest,free,up) ,(work,busy,half),(work,busy,done)}
L0,1 >< L0,2 >< L0,3 = (rest,free,up)

Act||h||n = {Tau_go,Tau_done,Tau_hit}
C1 U C2 U C3 ={x,y,z}
Tr||h||n ={t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12}

t1=((rest,free,up)    ---  (true,Tau_go,{x,y,z}) ----->  (work,busy,up))
t2=((work,busy,up)    ---  (x>=1,Tau_hit,{x})    ----->  (work,busy,half))
t3=((work,busy,half)  ---  (x>=1,Tau_hit,{x})    ----->  (work,busy,done))
t4=((work,busy,done)  ---  ((z>=10,y>=5,true) ,Tau_done,{})  ----->  (rest,free,up))
t5=((work,busy,done)  ---  (true,null,{})                    ----->  (rest,free,up))
t6=((work,busy,half)  ---  ((z>=10,y>=5,true),Tau_done,{})   ----->  (rest,free,half))
t7=((rest,free,half)  ---  (true,Tau_go,{x,y,z}) ----->  (work,busy,half))
t8=((work,busy,up)    ---  ((z>=10,y>=5,true),Tau_done,{})   ----->  (rest,free,up))
t9=((work,busy,done)  ---  ((z>=10,y>=5,true),Tau_done,{})   ----->  (rest,free,done))
t10=((work,busy,done) ---  (true,null,{})                    ----->  (work,busy,up))
t11=((rest,free,done) ---  (true,null,{})                    ----->  (rest,free,up))
t12=((rest,free,done) ---  (true,Tau_go,{x,y,z}) ----->  (work,busy,done))


Inv||h||n

Inv(rest,free,up)=true
Inv(work,busy,half)= z<=60
Inv(work,busy,done)= z<=60
Inv(work,busy,up)=z<=60
Inv(rest,free,half)=true
Inv(rest,free,done)=true
```

1.4 Exercício 4

```
N={in , out, d} | d in R0+

R:T={t1,t2,t3,t4}
t1={(off,t)-- d -->(off,t+d)}  |  t,d  >=0
t2={(on,t)-- d -->(on,t+d)}    |  t,d  >=0  and  t+d<=2
t3={(off,t)-- in -->(on,0)}    |  t >=0
t3={(on,t)-- out -->(off,t )}  |  1 =< t =< 2
```

Figura 1.3: Resolução

1.5 Exercício 5

```
R:  {}
        <0,press>, <3,press>,<4,press>
        <0,press>, <5,press>
```

Figura 1.4: Resolução

1.6 Exercício 6

```
Não, pq existem vários traços que num sistema podem ser executados e no outro não.
Por exemplo:
<1,t>,<x,t> | 1 =<x <=2 não pode ser executado no segundo sistema.

No entanto, no segundo sistema também é possivel acontecer o mesmo.
<1,t>,<x,t> | 0 =<x <1
```

Figura 1.5: Resolução

1.7 Exercício 7

```
R= { (s,{x-> d}), (v,{ x -> d }) | d E [0,10] } U
   { (t,{x-> d} ),(v,{ x -> d }) | d E [0,10] }


Sendo a1 uma ação e d um delay:

s --- a1 ---> t  => v --- a1 ---> v              AND  t R v

s --- d ---> t  => v --- d ---> v   | d E [2,10] AND t R v


Converso:

v --- a1 ---> v  => s --- a1 ---> t   AND  v R t

v --- d ---> v  => s --- d ---> t   |  d E [2,10] AND  v R t


---- Para a outra possivel ação:

t --- a1 ---> t  => v --- a1 ---> v              AND  t R v

t --- d ---> t  => v --- d ---> v   | d E [2,10] AND t R v

Converso:
É exatamente igual ao de cima.

Logo são timed-Bisimulation
-----------------------------------

Por exemplo :

s --- a1 ---> t  => v --- a1 ---> v   AND  t R v  = True

s --- d=3 ---> t  => v --- d=3 ---> v   AND  t R v = True

porque { (t,{x->3}), (v,{x->3}) } E R
```

Figura 1.6: Resolução