

LCTrabalho4 Exercicio2

January 17, 2021

1 Trabalho 4

1.1 Lógica Computacional 2020-2021

Trabalho realizado por:

- > 1. Paulo Costa - A87986
- > 2. André Araújo - A87987

1.1.1 Exercício 2

2. Considere um sistema híbrido formado por 4 autómatos híbridos: três navios (análogos aos do trabalho TP3) e um controlador. Neste sistema cada autômato desconhece o estado dos restantes e comunica com eles exclusivamente através de eventos.

A propriedade de segurança é a mesma da do trabalho TP2: ausência de colisões entre navios. Para isso a área de navegação é dividida em setores e o controlador assegura que, em qualquer instante, cada sector não contém mais do que um navio.

Assim

- a. Cada navio está, em cada estado, numa de três velocidades v possíveis: 10m/s (high) , 1m/s (low) e 0 (stop). As transições $\text{high} \leftrightarrow \text{low}$ têm uma duração mínima de 500 seg ; as transições $\text{low} \leftrightarrow \text{stop}$ têm uma duração mínima de 50 seg.
- b. Cada navio está, em cada estado, numa rota $r \in \{0..23\}$; cada valor de r identifica um ângulo múltiplo de 15° (também designado por **hora**).
- c. A área de navegação é dividida numa matriz $N \times N$ de setores quadrados com 1km de lado. Cada setor é identificado por um par de índices $0 \leq \text{linha}, \text{coluna} < N$. Cada navio está, em cada estado, num único setor.
- d. O estado do controlador incluiu o seu setor e a sua velocidade.

A navegação é determinada pelas seguintes regras:

- >a. Um navio só muda de rota ou velocidade quando muda de setor. \

- >b. Quando um navio entra ou sai de um setor emite um evento, que identifica o navio e os setores.
- >c. Se existir risco de dois navios estarem simultaneamente no mesmo setor, o controlador deve...
- >d. Dois navios em setores adjacentes estão ambos em velocidade low ou high .

Pretende-se verificar, dada uma determinada posição inicial dos três navios, a seguinte propriedade de segurança:

Em qualquer traço e em qualquer estado, nenhum setor contém mais do que um navio.

1.2 Resolução:

Para modelar o sistema híbrido definimos 4 automatos híbridos, um para cada barco e um para o controlador. >Os automatos dos barcos tem um modo que é definido como sendo um par $(Rota \in \{0..23\}, Velocidade \in (\text{high} \vee \text{low} \vee \text{stop}))$, o setor onde se encontra, a posição do barco e o tempo. >O controlador tem a informação relativa ao modo e ao setor de cada barco, tendo também em adição uma variável o tempo utilizada na transição de velocidades, para além do tempo do controlador.

Nas transições timed a variação do tempo das variáveis é igual o controlador fica igual e os barcos movimentam-se de acordo com o modo definido pelo o controlador e caso se verifique uma mudança de setor o barco muda o seu setor.

Nas transições untimed os barcos ficam na mesma posição e setor, a variação do tempo é 0, o controlador atualiza os setores dos barcos e caso um barco tenha mudado de setor muda a rota é atualizada, caso um barco esteja em perigo ou seja quando tenha um barco num dos setores adjacentes o barco diminui a velocidade e quando já esteja seguro o barco volta para HIGHT.

As transições untimed ocorrem quando um barco muda de setor.

Primeiramente, iremos passar a mostrar as nossas funções, cada uma delas com o comentário que, de facto, indica o objetivo de cada função ou de um aspeto a destacar no código utilizado.

```
[1]: from z3 import *
from random import choice, randint
from numpy import cos, sin, deg2rad
Velocidade, (STOP, LOW, HIGHT) = EnumSort('Velocidade', ('STOP', 'LOW', 'HIGHT'))

Rotas=[str(i) for i in range(24)]
lRotas=[i for i in range(0,360,15)]
Rota, Rotas = EnumSort('Velocidade', [str(i) for i in Rotas])
# Para auxiliar a mudar a rota e mover os barcos #
d={}
for i in range(24):
    d[Rotas[i]]={}
    d[Rotas[i]]['c']=cos(deg2rad(lRotas[i]))
    d[Rotas[i]]['s']=sin(deg2rad(lRotas[i]))
```

```
[2]: def declare(i):
    s = {}
```

```

# barco1 #
s['b1']={}

↳ s['b1']['m']=(Const('rota_barco1_'+str(i),Rota),Const('velocidade_barco1_'+str(i),Velocidad
s['b1']['s']=(Int('setor_b1_x_'+str(i)),Int('setor_b1_y_'+str(i)))
s['b1']['p']=(Real('x_barco1_'+str(i)),Real('y_barco1_'+str(i)))
s['b1']['t']=(Real('t_barco1_'+str(i)))
# barco2 #
s['b2']={}

↳ s['b2']['m']=(Const('rota_barco2_'+str(i),Rota),Const('velocidade_barco2_'+str(i),Velocidad
s['b2']['s']=(Int('setor_barco2_x_'+str(i)),Int('setor_barco2_y_'+str(i)))
s['b2']['p']=(Real('x_barco2_'+str(i)),Real('y_barco2_'+str(i)))
s['b2']['t']=(Real('t_barco2_'+str(i)))
# barco3 #
s['b3']={}

↳ s['b3']['m']=(Const('rota_barco3_'+str(i),Rota),Const('velocidade_barco3_'+str(i),Velocidad
s['b3']['s']=(Int('setor_barco3_x_'+str(i)),Int('setor_barco3_y_'+str(i)))
s['b3']['p']=(Real('x_barco3_'+str(i)),Real('y_barco3_'+str(i)))
s['b3']['t']=(Real('t_barco3_'+str(i)))
# controlador #
s['con']={}
s['con']['b1']={}

↳ s['con']['b1']['m']=(Const('c_rota_barco1_'+str(i),Rota),Const('c_velocidade_barco1_'+str(i
↳ s['con']['b1']['s']=(Int('c_setor_b1_x_'+str(i)),Int('c_setor_b1_y_'+str(i)))
s['con']['b1']['tt']=Real('c_barco1_t_'+str(i))
s['con']['b2']={}

↳ s['con']['b2']['m']=(Const('c_rota_barco2_'+str(i),Rota),Const('c_velocidade_barco2_'+str(i
↳ s['con']['b2']['s']=(Int('c_setor_b2_x_'+str(i)),Int('c_setor_b2_y_'+str(i)))
s['con']['b2']['tt']=Real('c_barco2_t_'+str(i))
s['con']['b3']={}

↳ s['con']['b3']['m']=(Const('c_rota_barco3_'+str(i),Rota),Const('c_velocidade_barco3_'+str(i
↳ s['con']['b3']['s']=(Int('c_setor_b3_x_'+str(i)),Int('c_setor_b3_y_'+str(i)))
s['con']['b3']['tt']=Real('c_barco3_t_'+str(i))
s['con']['t']=(Real('t_con_'+str(i)))

return s

def init(s):

```

```

# Inicia o barco 1 #
↳ b1=And(s['b1']['m'][0]==s['con']['b1']['m'][0],s['b1']['m'][1]==s['con']['b1']['m'][1],
s['b1']['s'][0]==randint(1,10),s['b1']['s'][1]==randint(1,10),
↳ s['b1']['p'][0]==s['b1']['s'][1]*1000+randint(0,1000),s['b1']['p'][1]==s['b1']['s'][1]*1000
s['b1']['t']==0)
# Inicia o barco 2 #
↳ b2=And(s['b2']['m'][0]==s['con']['b2']['m'][0],s['b2']['m'][1]==s['con']['b2']['m'][1],
s['b2']['s'][0]==randint(1,10),s['b2']['s'][1]==randint(1,10),
↳ s['b2']['p'][0]==s['b2']['s'][1]*1000+randint(0,1000),s['b2']['p'][1]==s['b2']['s'][1]*1000
s['b2']['t']==0)
# Inicia o barco 3 #
↳ b3=And(s['b3']['m'][0]==s['con']['b3']['m'][0],s['b3']['m'][1]==s['con']['b3']['m'][1],
s['b3']['s'][0]==randint(1,10),s['b3']['s'][1]==randint(1,10),
↳ s['b3']['p'][0]==s['b3']['s'][1]*1000+randint(0,1000),s['b3']['p'][1]==s['b3']['s'][1]*1000
s['b3']['t']==0)
# Inicia o controlador #
↳ c=And(s['con']['b1']['m'][0]==choice(Rotas),s['con']['b1']['m'][1]==HIGHT,s['con']['b1']['t']
↳ s['con']['b1']['s'][0]==s['b1']['s'][0],s['con']['b1']['s'][1]==s['b1']['s'][1],
↳ s['con']['b2']['m'][0]==choice(Rotas),s['con']['b2']['m'][1]==HIGHT,s['con']['b2']['tt']==0
↳ s['con']['b2']['s'][0]==s['b2']['s'][0],s['con']['b2']['s'][1]==s['b2']['s'][1],
↳ s['con']['b3']['m'][0]==choice(Rotas),s['con']['b3']['m'][1]==HIGHT,s['con']['b3']['tt']==0
↳ s['con']['b3']['s'][0]==s['b3']['s'][0],s['con']['b3']['s'][1]==s['b3']['s'][1],
s['con']['t']==0)

return And(b1,b2,b3,c)

```

[3]: # Função auxiliar que move a posição de cada barco #

```

def mover(s,p,b,v):
    return
↳ Or((And(s[b]['m'][0]==Rotas[a],p[b]['p'][0]==s[b]['p'][0]+d[Rotas[a]]['c']*v,
p[b]['p'][1]==s[b]['p'][1]+d[Rotas[a]]['s']*v)
for a in range(24)))

```

```
# Função que indica se existe perigo entre 2 barcos #
```

```
def danger(s,b1,b2,d):
    x=And(s['con'][b1]['s'][0]<=s['con'][b2]['s'][0]+d
    →,s['con'][b2]['s'][0]<=s['con'][b1]['s'][0]+d
    y=And(s['con'][b1]['s'][1]<=s['con'][b2]['s'][1]+d
    →,s['con'][b2]['s'][1]<=s['con'][b1]['s'][1]+d
    return And(x,y)
```

```
[4]: def trans(s,p):
    r=1
    # Auxiliar #
    # Modo igual #
    Im_b1=And(p['b1']['m'][0]==s['b1']['m'][0],p['b1']['m'][1]==s['b1']['m'][1])
    Im_b2=And(p['b2']['m'][0]==s['b2']['m'][0],p['b2']['m'][1]==s['b2']['m'][1])
    Im_b3=And(p['b3']['m'][0]==s['b3']['m'][0],p['b3']['m'][1]==s['b3']['m'][1])
    →Im_con_b1=And(p['con']['b1']['m'][0]==s['con']['b1']['m'][0],p['con']['b1']['m'][1]==s['con
    →Im_con_b2=And(p['con']['b2']['m'][0]==s['con']['b2']['m'][0],p['con']['b2']['m'][1]==s['con
    →Im_con_b3=And(p['con']['b3']['m'][0]==s['con']['b3']['m'][0],p['con']['b3']['m'][1]==s['con

    # Perigo #
    d_b1=Or(danger(s,'b1','b2',r),danger(s,'b1','b3',r))
    d_b2=Or(danger(s,'b2','b1',r),danger(s,'b2','b3',r))
    d_b3=Or(danger(s,'b3','b1',r),danger(s,'b3','b2',r))
    d_b1p=Or(danger(p,'b1','b2',r),danger(p,'b1','b3',r))
    d_b2p=Or(danger(p,'b2','b1',r),danger(p,'b2','b3',r))
    d_b3p=Or(danger(p,'b3','b1',r),danger(p,'b3','b2',r))
    # Seguro #
    sa_b1=Not(d_b1)
    sa_b2=Not(d_b2)
    sa_b3=Not(d_b3)
    sa_b1p=Not(d_b1p)
    sa_b2p=Not(d_b2p)
    sa_b3p=Not(d_b3p)

    # Posicao igual #
    Ip_b1=And(p['b1']['p'][0]==s['b1']['p'][0],p['b1']['p'][1]==s['b1']['p'][1])
    Ip_b2=And(p['b2']['p'][0]==s['b2']['p'][0],p['b2']['p'][1]==s['b2']['p'][1])
    Ip_b3=And(p['b3']['p'][0]==s['b3']['p'][0],p['b3']['p'][1]==s['b3']['p'][1])

    # setor igual #
    Is_b1=And(p['b1']['s'][0]==s['b1']['s'][0],p['b1']['s'][1]==s['b1']['s'][1])
    Is_b2=And(p['b2']['s'][0]==s['b2']['s'][0],p['b2']['s'][1]==s['b2']['s'][1])
    Is_b3=And(p['b3']['s'][0]==s['b3']['s'][0],p['b3']['s'][1]==s['b3']['s'][1])
```

```

└─
→ Is_con_b1=And(p['con']['b1']['s'][0]==s['con']['b1']['s'][0],p['con']['b1']['s'][1]==s['con
└─
→ Is_con_b2=And(p['con']['b2']['s'][0]==s['con']['b2']['s'][0],p['con']['b2']['s'][1]==s['con
└─
→ Is_con_b3=And(p['con']['b3']['s'][0]==s['con']['b3']['s'][0],p['con']['b3']['s'][1]==s['con

# igualar a variacao de tempo #
vt=And(p['con']['t']-s['con']['t']==p['b1']['t']-s['b1']['t'],
        p['con']['t']-s['con']['t']==p['b2']['t']-s['b2']['t'],
        p['con']['t']-s['con']['t']==p['b3']['t']-s['b3']['t'])

└─
→ vt_vb1=(p['con']['t']-s['con']['t']==p['con']['b1']['tt']-s['con']['b1']['tt'])
└─
→ vt_vb2=(p['con']['t']-s['con']['t']==p['con']['b2']['tt']-s['con']['b2']['tt'])
└─
→ vt_vb3=(p['con']['t']-s['con']['t']==p['con']['b3']['tt']-s['con']['b3']['tt'])

# ver se o setor esta correto #
# setor x esta igual #
└─
→ sc_b1x=And(p['b1']['p'][0]>=s['b1']['s'][0]*1000,p['b1']['p'][0]<s['b1']['s'][0]*1000+1000,
              p['b1']['s'][0]==s['b1']['s'][0])
└─
→ sc_b2x=And(p['b2']['p'][0]>=s['b2']['s'][0]*1000,p['b2']['p'][0]<s['b2']['s'][0]*1000+1000,
              p['b2']['s'][0]==s['b2']['s'][0])
└─
→ sc_b3x=And(p['b3']['p'][0]>=s['b3']['s'][0]*1000,p['b3']['p'][0]<s['b3']['s'][0]*1000+1000,
              p['b3']['s'][0]==s['b3']['s'][0])
# setor x menor #
└─
→ sm_b1x=And(p['b1']['p'][0]<s['b1']['s'][0]*1000,p['b1']['s'][0]==s['b1']['s'][0]-1)
└─
→ sm_b2x=And(p['b2']['p'][0]<s['b2']['s'][0]*1000,p['b2']['s'][0]==s['b2']['s'][0]-1)
└─
→ sm_b3x=And(p['b3']['p'][0]<s['b3']['s'][0]*1000,p['b3']['s'][0]==s['b3']['s'][0]-1)
# setor x maior #
└─
→ sM_b1x=And(p['b1']['p'][0]>=s['b1']['s'][0]*1000+1000,p['b1']['s'][0]==s['b1']['s'][0]+1)
└─
→ sM_b2x=And(p['b2']['p'][0]>=s['b2']['s'][0]*1000+1000,p['b2']['s'][0]==s['b2']['s'][0]+1)
└─
→ sM_b3x=And(p['b2']['p'][0]>=s['b3']['s'][0]*1000+1000,p['b3']['s'][0]==s['b3']['s'][0]+1)
# setor y esta igual #
└─
→ sc_b1y=And(p['b1']['p'][1]>=s['b1']['s'][1]*1000,p['b1']['p'][0]<s['b1']['s'][1]*1000+1000,

```

```

        p['b1']['s'][1]==s['b1']['s'][1])
    □
    ↪sc_b2y=And(p['b2']['p'][1]>=s['b2']['s'][1]*1000,p['b2']['p'][0]<s['b2']['s'][1]*1000+1000,
        p['b2']['s'][1]==s['b2']['s'][1])
    □
    ↪sc_b3y=And(p['b3']['p'][1]>=s['b3']['s'][1]*1000,p['b3']['p'][0]<s['b3']['s'][1]*1000+1000,
        p['b3']['s'][1]==s['b3']['s'][1])
    # setor y menor #
    □
    ↪sm_b1y=And(p['b1']['p'][1]<s['b1']['s'][1]*1000,p['b1']['s'][1]==s['b1']['s'][1]-1)
    □
    ↪sm_b2y=And(p['b2']['p'][1]<s['b2']['s'][1]*1000,p['b2']['s'][1]==s['b2']['s'][1]-1)
    □
    ↪sm_b3y=And(p['b3']['p'][1]<s['b3']['s'][1]*1000,p['b3']['s'][1]==s['b3']['s'][1]-1)
    # setor y maior #
    □
    ↪sM_b1y=And(p['b1']['p'][1]>=s['b1']['s'][1]*1000+1000,p['b1']['s'][1]==s['b1']['s'][1]+1)
    □
    ↪sM_b2y=And(p['b2']['p'][1]>=s['b2']['s'][1]*1000+1000,p['b2']['s'][1]==s['b2']['s'][1]+1)
    □
    ↪sM_b3y=And(p['b2']['p'][1]>=s['b3']['s'][1]*1000+1000,p['b3']['s'][1]==s['b3']['s'][1]+1)
    # verifica qual é o setor apos mover o barco #
    s_b1=And(Or(sc_b1x,sm_b1x,sM_b1x),Or(sc_b1y,sm_b1y,sM_b1y))
    s_b2=And(Or(sc_b2x,sm_b2x,sM_b2x),Or(sc_b2y,sm_b2y,sM_b2y))
    s_b3=And(Or(sc_b3x,sm_b3x,sM_b3x),Or(sc_b3y,sm_b3y,sM_b3y))

    # Untimed #

    # setor errado #
    □
    ↪s_err=Or(Not(s['b1']['s'][0]==s['con']['b1']['s'][0]),Not(s['b2']['s'][0]==s['con']['b2']['s'][0]),
    □
    ↪Not(s['b3']['s'][0]==s['con']['b3']['s'][0]),Not(s['b1']['s'][1]==s['con']['b1']['s'][1]),
    □
    ↪Not(s['b2']['s'][1]==s['con']['b2']['s'][1]),Not(s['b3']['s'][1]==s['con']['b3']['s'][1]))
    # salvo m==LOW or m==STOP #
    □
    ↪t_b1S=And(sa_b1,Or(And(s['con']['b1']['m'][1]==LOW,s['con']['b1']['tt']>=500),
    □
    ↪And(s['con']['b1']['m'][1]==STOP,s['con']['b1']['tt']>=50)))
    □
    ↪t_b2S=And(sa_b2,Or(And(s['con']['b2']['m'][1]==LOW,s['con']['b2']['tt']>=500),
    □
    ↪And(s['con']['b2']['m'][1]==STOP,s['con']['b2']['tt']>=50)))
    □
    ↪t_b3S=And(sa_b3,Or(And(s['con']['b3']['m'][1]==LOW,s['con']['b3']['tt']>=500),

```

```

    ↪And(s['con']['b3']['m'][1]==STOP,s['con']['b3']['tt']>=50)))
    # perigo m==HIGHT #
    t_b1D=And(d_b1,s['con']['b1']['m'][1]==HIGHT,s['con']['b1']['tt']>=500)
    t_b2D=And(d_b2,s['con']['b2']['m'][1]==HIGHT,s['con']['b2']['tt']>=500)
    t_b3D=And(d_b3,s['con']['b3']['m'][1]==HIGHT,s['con']['b3']['tt']>=500)
    # Condição para untimed # sb!=cSb perigo sem tempo de transição time e
    ↪safe sem tempo de transição #
    u=Or(t_b1S,t_b2S,t_b3S,t_b1D,t_b2D,t_b3D)
    # LOW To HIGHT #

    ↪b1_LtH=And(s['con']['b1']['m'][1]==LOW,Ip_b1,Is_b1,s['b1']['s'][0]==p['con']['b1']['s'][0],
    ↪s['b1']['s'][1]==p['con']['b1']['s'][1],p['con']['b1']['m'][1]==HIGHT,p['b1']['m'][1]==s['b1']
    p['b1']['m'][0]==p['con']['b1']['m'][0],p['con']['b1']['tt']==0,
    sa_b1p,s['con']['b1']['tt']>=500,

    ↪Or(And(Not(And(s['b1']['s'][0]==p['con']['b1']['s'][0],s['b1']['s'][1]==p['con']
    Or([p['con']['b1']['m'][0]==Rotas[i] for i in
    ↪range(24)]))),

    ↪And(s['b1']['s'][0]==p['con']['b1']['s'][0],s['b1']['s'][1]==p['con']['b1']['s'][1],
    p['b1']['m'][0]==s['b1']['m'][0]))

    ↪b2_LtH=And(s['con']['b2']['m'][1]==LOW,Ip_b2,Is_b2,s['b2']['s'][0]==p['con']['b2']['s'][0],
    ↪s['b2']['s'][1]==p['con']['b2']['s'][1],p['con']['b2']['m'][1]==HIGHT,p['b2']['m'][1]==s['b2']
    p['b2']['m'][0]==p['con']['b2']['m'][0],p['con']['b2']['tt']==0,
    sa_b2p,s['con']['b2']['tt']>=500,

    ↪Or(And(Not(And(s['b2']['s'][0]==p['con']['b2']['s'][0],s['b2']['s'][1]==p['con']
    Or([p['con']['b2']['m'][0]==Rotas[i] for i in
    ↪range(24)]))),

    ↪And(s['b2']['s'][0]==p['con']['b2']['s'][0],s['b2']['s'][1]==p['con']['b2']['s'][1],
    p['b2']['m'][0]==s['b2']['m'][0]))

    ↪b3_LtH=And(s['con']['b3']['m'][1]==LOW,Ip_b3,Is_b3,s['b3']['s'][0]==p['con']['b3']['s'][0],
    ↪s['b3']['s'][1]==p['con']['b3']['s'][1],p['con']['b3']['m'][1]==HIGHT,p['b3']['m'][1]==s['b3']
    p['b3']['m'][0]==p['con']['b3']['m'][0],p['con']['b3']['tt']==0,
    sa_b3p,s['con']['b3']['tt']>=500,

    ↪Or(And(Not(And(s['b3']['s'][0]==p['con']['b3']['s'][0],s['b3']['s'][1]==p['con']
    Or([p['con']['b3']['m'][0]==Rotas[i] for i in
    ↪range(24)]))),

```



```

    ↪And(s['b3']['s'][0]==p['con']['b3']['s'][0],s['b3']['s'][1]==p['con']['b3']['s'][1],
        p['b3']['m'][0]==s['b3']['m'][0]))
    # HIGHT To LOW #
    ↪b1_HtL=And(s['con']['b1']['m'][1]==HIGHT,Ip_b1,Is_b1,s['b1']['s'][0]==p['con']['b1']['s'][0],
        ↪s['b1']['s'][1]==p['con']['b1']['s'][1],p['con']['b1']['m'][1]==LOW,p['b1']['m'][1]==s['b1']
            p['b1']['m'][0]==p['con']['b1']['m'][0],p['con']['b1']['tt']==0,
            d_b1p,s['con']['b1']['tt']>=500,
        ↪Or(And(Not(And(s['b1']['s'][0]==p['con']['b1']['s'][0],s['b1']['s'][1]==p['con']['b1']['s']
            Or([p['con']['b1']['m'][0]==Rotas[i] for i in
        ↪range(24)]))),
        ↪And(s['b1']['s'][0]==p['con']['b1']['s'][0],s['b1']['s'][1]==p['con']['b1']['s'][1],
            p['b1']['m'][0]==s['b1']['m'][0]))
    ↪b2_HtL=And(s['con']['b2']['m'][1]==HIGHT,Ip_b2,Is_b2,s['b2']['s'][0]==p['con']['b2']['s'][0],
        ↪s['b2']['s'][1]==p['con']['b2']['s'][1],p['con']['b2']['m'][1]==LOW,p['b2']['m'][1]==s['b2']
            p['b2']['m'][0]==p['con']['b2']['m'][0],p['con']['b2']['tt']==0,
            d_b2p,s['con']['b2']['tt']>=500,
        ↪Or(And(Not(And(s['b2']['s'][0]==p['con']['b2']['s'][0],s['b2']['s'][1]==p['con']['b2']['s']
            Or([p['con']['b2']['m'][0]==Rotas[i] for i in
        ↪range(24)]))),
        ↪And(s['b2']['s'][0]==p['con']['b2']['s'][0],s['b2']['s'][1]==p['con']['b2']['s'][1],
            p['b2']['m'][0]==s['b2']['m'][0]))
    ↪b3_HtL=And(s['con']['b3']['m'][1]==HIGHT,Ip_b3,Is_b3,s['b3']['s'][0]==p['con']['b3']['s'][0],
        ↪s['b3']['s'][1]==p['con']['b3']['s'][1],p['con']['b3']['m'][1]==LOW,p['b3']['m'][1]==s['b3']
            p['b3']['m'][0]==p['con']['b3']['m'][0],p['con']['b3']['tt']==0,
            d_b3p,s['con']['b3']['tt']>=500,
        ↪Or(And(Not(And(s['b3']['s'][0]==p['con']['b3']['s'][0],s['b3']['s'][1]==p['con']['b3']['s']
            Or([p['con']['b3']['m'][0]==Rotas[i] for i in
        ↪range(24)]))),
        ↪And(s['b3']['s'][0]==p['con']['b3']['s'][0],s['b3']['s'][1]==p['con']['b3']['s'][1],
            p['b3']['m'][0]==s['b3']['m'][0]))
    # LOW To STOP #
    ↪b1_LtS=And(s['con']['b1']['m'][1]==LOW,Ip_b1,Is_b1,s['b1']['s'][0]==p['con']['b1']['s'][0],

```

```

└─→s['b1']['s'][1]==p['con']['b1']['s'][1],p['con']['b1']['m'][1]==STOP,p['b1']['m'][1]==s['b1']
    p['b1']['m'][0]==p['con']['b1']['m'][0],p['con']['b1']['tt']==0,
    d_b1p,s['con']['b1']['tt']>=50,
└─→Or(And(Not(And(s['b1']['s'][0]==p['con']['b1']['s'][0],s['b1']['s'][1]==p['con']['b1']['s']
    Or([p['con']['b1']['m'][0]==Rotas[i] for i in
└─→range(24)]))),
└─→And(s['b1']['s'][0]==p['con']['b1']['s'][0],s['b1']['s'][1]==p['con']['b1']['s'][1],
    p['b1']['m'][0]==s['b1']['m'][0]))))
└─→b2_LtS=And(s['con']['b2']['m'][1]==LOW,Ip_b2,Is_b2,s['b2']['s'][0]==p['con']['b2']['s'][0],
└─→s['b2']['s'][1]==p['con']['b2']['s'][1],p['con']['b2']['m'][1]==STOP,p['b2']['m'][1]==s['b2']
    p['b2']['m'][0]==p['con']['b2']['m'][0],p['con']['b2']['tt']==0,
    d_b2p,s['con']['b2']['tt']>=50,
└─→Or(And(Not(And(s['b2']['s'][0]==p['con']['b2']['s'][0],s['b2']['s'][1]==p['con']['b2']['s']
    Or([p['con']['b2']['m'][0]==Rotas[i] for i in
└─→range(24)]))),
└─→And(s['b2']['s'][0]==p['con']['b2']['s'][0],s['b2']['s'][1]==p['con']['b2']['s'][1],
    p['b2']['m'][0]==s['b2']['m'][0]))))
└─→b3_LtS=And(s['con']['b3']['m'][1]==LOW,Ip_b3,Is_b3,s['b3']['s'][0]==p['con']['b3']['s'][0],
└─→s['b3']['s'][1]==p['con']['b3']['s'][1],p['con']['b3']['m'][1]==STOP,p['b3']['m'][1]==s['b3']
    p['b3']['m'][0]==p['con']['b3']['m'][0],p['con']['b3']['tt']==0,
    d_b3p,s['con']['b3']['tt']>=50,
└─→Or(And(Not(And(s['b3']['s'][0]==p['con']['b3']['s'][0],s['b3']['s'][1]==p['con']['b3']['s']
    Or([p['con']['b3']['m'][0]==Rotas[i] for i in
└─→range(24)]))),
└─→And(s['b3']['s'][0]==p['con']['b3']['s'][0],s['b3']['s'][1]==p['con']['b3']['s'][1],
    p['b3']['m'][0]==s['b3']['m'][0]))))
    # STOP To LOW #
└─→b1_StL=And(s['con']['b1']['m'][1]==STOP,Ip_b1,Is_b1,s['b1']['s'][0]==p['con']['b1']['s'][0]
└─→s['b1']['s'][1]==p['con']['b1']['s'][1],p['con']['b1']['m'][1]==LOW,p['b1']['m'][1]==s['b1']
    p['b1']['m'][0]==p['con']['b1']['m'][0],p['con']['b1']['tt']==0,
    sa_b1p,s['con']['b1']['tt']>=50,
└─→Or(And(Not(And(s['b1']['s'][0]==p['con']['b1']['s'][0],s['b1']['s'][1]==p['con']['b1']['s']

```

```

Or([p['con']['b1']['m'][0]==Rotas[i] for i in
↪range(24)])),
↪
↪And(s['b1']['s'][0]==p['con']['b1']['s'][0],s['b1']['s'][1]==p['con']['b1']['s'][1],
↪p['b1']['m'][0]==s['b1']['m'][0]))
↪
↪b2_StL=And(s['con']['b2']['m'][1]==STOP,Ip_b2,Is_b2,s['b2']['s'][0]==p['con']['b2']['s'][0]
↪s['b2']['s'][1]==p['con']['b2']['s'][1],p['con']['b2']['m'][1]==LOW,p['b2']['m'][1]==s['b2']
↪p['b2']['m'][0]==p['con']['b2']['m'][0],p['con']['b2']['tt']==0,
↪sa_b2p,s['con']['b2']['tt']>=50,
↪
↪Or(And(Not(And(s['b2']['s'][0]==p['con']['b2']['s'][0],s['b2']['s'][1]==p['con']['b2']['s']
↪Or([p['con']['b2']['m'][0]==Rotas[i] for i in
↪range(24)])),
↪
↪And(s['b2']['s'][0]==p['con']['b2']['s'][0],s['b2']['s'][1]==p['con']['b2']['s'][1],
↪p['b2']['m'][0]==s['b2']['m'][0]))
↪
↪b3_StL=And(s['con']['b3']['m'][1]==STOP,Ip_b3,Is_b3,s['b3']['s'][0]==p['con']['b3']['s'][0]
↪s['b3']['s'][1]==p['con']['b3']['s'][1],p['con']['b3']['m'][1]==LOW,p['b3']['m'][1]==s['b3']
↪p['b3']['m'][0]==p['con']['b3']['m'][0],p['con']['b3']['tt']==0,
↪sa_b3p,s['con']['b3']['tt']>=50,
↪
↪Or(And(Not(And(s['b3']['s'][0]==p['con']['b3']['s'][0],s['b3']['s'][1]==p['con']['b3']['s']
↪Or([p['con']['b3']['m'][0]==Rotas[i] for i in
↪range(24)])),
↪
↪And(s['b3']['s'][0]==p['con']['b3']['s'][0],s['b3']['s'][1]==p['con']['b3']['s'][1],
↪p['b3']['m'][0]==s['b3']['m'][0]))

# HIGHT To HIGHT #
↪
↪b1_HtH=And(s['con']['b1']['m'][1]==HIGHT,Ip_b1,Is_b1,s['b1']['s'][0]==p['con']['b1']['s'][0]
↪s['b1']['s'][1]==p['con']['b1']['s'][1],p['con']['b1']['m'][1]==HIGHT,p['b1']['m'][1]==s['b
↪
↪p['b1']['m'][0]==p['con']['b1']['m'][0],p['con']['b1']['tt']==s['con']['b1']['tt'],
↪Or(sa_b1p,And(d_b1p,s['con']['b1']['tt']<500)),
↪
↪Or(And(Not(And(s['b1']['s'][0]==p['con']['b1']['s'][0],s['b1']['s'][1]==p['con']['b1']['s']
↪Or([p['con']['b1']['m'][0]==Rotas[i] for i in
↪range(24)])),
↪
↪And(s['b1']['s'][0]==p['con']['b1']['s'][0],s['b1']['s'][1]==p['con']['b1']['s'][1],

```

```

        p['b1']['m'][0]==s['b1']['m'][0]))
    ↪
    ↪b2_HtH=And(s['con']['b2']['m'][1]==HIGHT,Ip_b2,Is_b2,s['b2']['s'][0]==p['con']['b2']['s'][0],
    ↪
    ↪s['b2']['s'][1]==p['con']['b2']['s'][1],p['con']['b2']['m'][1]==HIGHT,p['b2']['m'][1]==s['b2']['m'][1],
    ↪
    ↪p['b2']['m'][0]==p['con']['b2']['m'][0],p['con']['b2']['tt']==s['con']['b2']['tt'],
        Or(sa_b2p,And(d_b2p,s['con']['b2']['tt']<500)),
    ↪
    ↪Or(And(Not(And(s['b2']['s'][0]==p['con']['b2']['s'][0],s['b2']['s'][1]==p['con']['b2']['s'][1],
        Or([p['con']['b2']['m'][0]==Rotas[i] for i in
    ↪range(24)]))),
    ↪
    ↪And(s['b2']['s'][0]==p['con']['b2']['s'][0],s['b2']['s'][1]==p['con']['b2']['s'][1],
        p['b2']['m'][0]==s['b2']['m'][0]))
    ↪
    ↪b3_HtH=And(s['con']['b3']['m'][1]==HIGHT,Ip_b3,Is_b3,s['b3']['s'][0]==p['con']['b3']['s'][0],
    ↪
    ↪s['b3']['s'][1]==p['con']['b3']['s'][1],p['con']['b3']['m'][1]==HIGHT,p['b3']['m'][1]==s['b3']['m'][1],
    ↪
    ↪p['b3']['m'][0]==p['con']['b3']['m'][0],p['con']['b3']['tt']==s['con']['b3']['tt'],
        Or(sa_b3p,And(d_b3p,s['con']['b3']['tt']<500)),
    ↪
    ↪Or(And(Not(And(s['b3']['s'][0]==p['con']['b3']['s'][0],s['b3']['s'][1]==p['con']['b3']['s'][1],
        Or([p['con']['b3']['m'][0]==Rotas[i] for i in
    ↪range(24)]))),
    ↪
    ↪And(s['b3']['s'][0]==p['con']['b3']['s'][0],s['b3']['s'][1]==p['con']['b3']['s'][1],
        p['b3']['m'][0]==s['b3']['m'][0]))
    ↪
    ↪# LOW To LOW #
    ↪
    ↪b1_LtL=And(s['con']['b1']['m'][1]==LOW,Ip_b1,Is_b1,s['b1']['s'][0]==p['con']['b1']['s'][0],
    ↪
    ↪s['b1']['s'][1]==p['con']['b1']['s'][1],p['con']['b1']['m'][1]==LOW,p['b1']['m'][1]==s['b1']['m'][1],
    ↪
    ↪p['b1']['m'][0]==p['con']['b1']['m'][0],p['con']['b1']['tt']==s['con']['b1']['tt'],
        Or(d_b1p,And(sa_b1p,s['con']['b1']['tt']<500)),
    ↪
    ↪Or(And(Not(And(s['b1']['s'][0]==p['con']['b1']['s'][0],s['b1']['s'][1]==p['con']['b1']['s'][1],
        Or([p['con']['b1']['m'][0]==Rotas[i] for i in
    ↪range(24)]))),
    ↪
    ↪And(s['b1']['s'][0]==p['con']['b1']['s'][0],s['b1']['s'][1]==p['con']['b1']['s'][1],
        p['b1']['m'][0]==s['b1']['m'][0]))
    ↪
    ↪b2_LtL=And(s['con']['b2']['m'][1]==LOW,Ip_b2,Is_b2,s['b2']['s'][0]==p['con']['b2']['s'][0],

```

```

    ↪s['b2']['s'][1]==p['con']['b2']['s'][1],p['con']['b2']['m'][1]==LOW,p['b2']['m'][1]==s['b2']
    ↪p['b2']['m'][0]==p['con']['b2']['m'][0],p['con']['b2']['tt']==s['con']['b2']['tt'],
        Or(d_b2p,And(sa_b2p,s['con']['b2']['tt']<500)),
    ↪Or(And(Not(And(s['b2']['s'][0]==p['con']['b2']['s'][0],s['b2']['s'][1]==p['con']['b2']['s']
        Or([p['con']['b2']['m'][0]==Rotas[i] for i in
    ↪range(24)]))),
    ↪And(s['b2']['s'][0]==p['con']['b2']['s'][0],s['b2']['s'][1]==p['con']['b2']['s'][1],
        p['b2']['m'][0]==s['b2']['m'][0]))
    ↪b3_LtL=And(s['con']['b3']['m'][1]==LOW,Ip_b3,Is_b3,s['b3']['s'][0]==p['con']['b3']['s'][0],
    ↪s['b3']['s'][1]==p['con']['b3']['s'][1],p['con']['b3']['m'][1]==LOW,p['b3']['m'][1]==s['b3']
    ↪p['b3']['m'][0]==p['con']['b3']['m'][0],p['con']['b3']['tt']==s['con']['b3']['tt'],
        Or(d_b3p,And(sa_b3p,s['con']['b3']['tt']<500)),
    ↪Or(And(Not(And(s['b3']['s'][0]==p['con']['b3']['s'][0],s['b3']['s'][1]==p['con']['b3']['s']
        Or([p['con']['b3']['m'][0]==Rotas[i] for i in
    ↪range(24)]))),
    ↪And(s['b3']['s'][0]==p['con']['b3']['s'][0],s['b3']['s'][1]==p['con']['b3']['s'][1],
        p['b3']['m'][0]==s['b3']['m'][0]))
    # STOP To STOP #
    ↪b1_StS=And(s['con']['b1']['m'][1]==STOP,Ip_b1,Is_b1,s['b1']['s'][0]==p['con']['b1']['s'][0]
    ↪s['b1']['s'][1]==p['con']['b1']['s'][1],p['con']['b1']['m'][1]==STOP,p['b1']['m'][1]==s['b1']
    ↪p['b1']['m'][0]==p['con']['b1']['m'][0],p['con']['b1']['tt']==s['con']['b1']['tt'],
        Or(d_b1p,And(sa_b1p,s['con']['b1']['tt']<50)),
    ↪Or(And(Not(And(s['b1']['s'][0]==p['con']['b1']['s'][0],s['b1']['s'][1]==p['con']['b1']['s']
        Or([p['con']['b1']['m'][0]==Rotas[i] for i in
    ↪range(24)]))),
    ↪And(s['b1']['s'][0]==p['con']['b1']['s'][0],s['b1']['s'][1]==p['con']['b1']['s'][1],
        p['b1']['m'][0]==s['b1']['m'][0]))
    ↪b2_StS=And(s['con']['b2']['m'][1]==STOP,Ip_b2,Is_b2,s['b2']['s'][0]==p['con']['b2']['s'][0]
    ↪s['b2']['s'][1]==p['con']['b2']['s'][1],p['con']['b2']['m'][1]==STOP,p['b2']['m'][1]==s['b2']

```

```

    ↪p['b2']['m'][0]==p['con']['b2']['m'][0],p['con']['b2']['tt']==s['con']['b2']['tt'],
        Or(d_b2p,And(sa_b2p,s['con']['b2']['tt']<50)),
    ↪Or(And(Not(And(s['b2']['s'][0]==p['con']['b2']['s'][0],s['b2']['s'][1]==p['con']['b2']['s'][1],
        Or([p['con']['b2']['m'][0]==Rotas[i] for i in
    ↪range(24)]))),
    ↪And(s['b2']['s'][0]==p['con']['b2']['s'][0],s['b2']['s'][1]==p['con']['b2']['s'][1],
        p['b2']['m'][0]==s['b2']['m'][0]))))
    ↪b3_StS=And(s['con']['b3']['m'][1]==STOP,Ip_b3,Is_b3,s['b3']['s'][0]==p['con']['b3']['s'][0]
    ↪s['b3']['s'][1]==p['con']['b3']['s'][1],p['con']['b3']['m'][1]==STOP,p['b3']['m'][1]==s['b3']
    ↪p['b3']['m'][0]==p['con']['b3']['m'][0],p['con']['b3']['tt']==s['con']['b3']['tt'],
        Or(d_b3p,And(sa_b3p,s['con']['b3']['tt']<50)),
    ↪Or(And(Not(And(s['b3']['s'][0]==p['con']['b3']['s'][0],s['b3']['s'][1]==p['con']['b3']['s'][1],
        Or([p['con']['b3']['m'][0]==Rotas[i] for i in
    ↪range(24)]))),
    ↪And(s['b3']['s'][0]==p['con']['b3']['s'][0],s['b3']['s'][1]==p['con']['b3']['s'][1],
        p['b3']['m'][0]==s['b3']['m'][0]))))
    # Transição de cada barco #
    b1_U=Or(b1_StS,b1_LtL,b1_HtH,b1_HtL,b1_LtH,b1_LtS,b1_StL)
    b2_U=Or(b2_StS,b2_LtL,b2_HtH,b2_HtL,b2_LtH,b2_LtS,b2_StL)
    b3_U=Or(b3_StS,b3_LtL,b3_HtH,b3_HtL,b3_LtH,b3_LtS,b3_StL)

    # Transição #
    untimed=And(u,vt,p['con']['t']==s['con']['t'],b1_U,b2_U,b3_U)

    # Timed #
    # Transição HIGHT de cada barco #
    ↪b1_H=And(s['b1']['m'][1]==HIGHT,Im_b1,mover(s,p,'b1',10),p['b1']['t']>s['b1']['t'],s_b1)
    ↪b2_H=And(s['b2']['m'][1]==HIGHT,Im_b1,mover(s,p,'b2',10),p['b2']['t']>s['b2']['t'],s_b2)
    ↪b3_H=And(s['b3']['m'][1]==HIGHT,Im_b1,mover(s,p,'b3',10),p['b3']['t']>s['b3']['t'],s_b3)
    # Transição LOW de cada barco #
    ↪b1_L=And(s['b1']['m'][1]==LOW,Im_b1,mover(s,p,'b1',1),p['b1']['t']>s['b1']['t'],s_b1)
    ↪b2_L=And(s['b2']['m'][1]==LOW,Im_b1,mover(s,p,'b2',1),p['b2']['t']>s['b2']['t'],s_b2)

```

```

└─
→ b3_L=And(s['b3']['m'][1]==LOW,Im_b1,mover(s,p,'b3',1),p['b3']['t']>s['b3']['t'],s_b3)
    # Transição STOP de cada barco #
    b1_S=And(s['b1']['m'][1]==STOP,Im_b1,Is_b1,Ip_b1,p['b1']['t']>s['b1']['t'])
    b2_S=And(s['b2']['m'][1]==STOP,Im_b2,Is_b2,Ip_b2,p['b2']['t']>s['b2']['t'])
    b3_S=And(s['b3']['m'][1]==STOP,Im_b3,Is_b3,Ip_b3,p['b3']['t']>s['b3']['t'])
    # Transição de cada barco #
    b1_T=Or(b1_H,b1_L,b1_S)
    b2_T=Or(b2_H,b2_L,b2_S)
    b3_T=Or(b3_H,b3_L,b3_S)
└─
→ con_T=And(Im_con_b1,Im_con_b2,Im_con_b3,Is_con_b1,Is_con_b2,Is_con_b3,vt_vb1,vt_vb2,vt_vb3)
    # Transição #
    timed=And(b1_T,b2_T,b3_T,con_T,vt,p['con']['t']>s['con']['t'],Not(u))
    return Or(timed,untimed)

```

```

[6]: def gera_traco(declare,init,trans,k):
    s = Solver()
    state=[declare(i) for i in range(k)]
    s.add(init(state[0]))
    for i in range(k-1):
        s.add(trans(state[i],state[i+1]))
    if s.check()==sat:
        m=s.model()
        print(m)
    else:
        print('unsat')

gera_traco(declare,init,trans,2)

```

```

[setor_barco3_y_1 = 8,
 c_barco1_t1 = 1,
 setor_barco3_x_1 = 7,
 y_barco3_1 = 962176476193621849/1250000000000000,
 x_barco3_1 = 7655340741737109317/10000000000000000,
 velocidade_barco1_1 = HIGHT,
 setor_barco2_y_1 = 10,
 setor_barco2_x_1 = 9,
 y_barco2_1 = 53897940952255126037/5000000000000000,
 x_barco2_1 = 10996659258262890683/10000000000000000,
 setor_b1_y_1 = 2,
 setor_b1_x_1 = 9,
 y_barco1_1 =
287900000000000000012246467991473533/100000000000000000000000000000000,
 x_barco1_1 = 2175,
 t_barco3_1 = 1,
 t_barco2_1 = 1,
 t_barco1_1 = 1,

```

```

c_barco3_t1 = 1,
c_barco2_t1 = 1,
t_con_1 = 1,
c_setor_b3_y_1 = 7,
c_setor_b3_x_1 = 6,
c_setor_b2_y_1 = 10,
c_setor_b2_x_1 = 8,
c_setor_b1_y_1 = 2,
c_setor_b1_x_1 = 10,
c_velocidade_barco3_1 = HIGHT,
c_rota_barco3_1 = 13,
c_velocidade_barco2_1 = HIGHT,
c_rota_barco2_1 = 1,
c_velocidade_barco1_1 = HIGHT,
c_rota_barco1_1 = 12,
rota_barco1_1 = 12,
t_con_0 = 0,
c_setor_b3_y_0 = 7,
c_setor_b3_x_0 = 6,
c_barco3_t0 = 0,
c_velocidade_barco3_0 = HIGHT,
c_rota_barco3_0 = 13,
c_setor_b2_y_0 = 10,
c_setor_b2_x_0 = 8,
c_barco2_t0 = 0,
c_velocidade_barco2_0 = HIGHT,
c_rota_barco2_0 = 1,
c_setor_b1_y_0 = 2,
c_setor_b1_x_0 = 10,
c_barco1_t0 = 0,
c_velocidade_barco1_0 = HIGHT,
c_rota_barco1_0 = 12,
t_barco3_0 = 0,
y_barco3_0 = 7700,
x_barco3_0 = 7665,
setor_barco3_y_0 = 7,
setor_barco3_x_0 = 6,
velocidade_barco3_0 = HIGHT,
rota_barco3_0 = 13,
t_barco2_0 = 0,
y_barco2_0 = 10777,
x_barco2_0 = 10987,
setor_barco2_y_0 = 10,
setor_barco2_x_0 = 8,
velocidade_barco2_0 = HIGHT,
rota_barco2_0 = 1,
t_barco1_0 = 0,
y_barco1_0 = 2879,

```



```

x_barco1_0 = 2185,
setor_b1_y_0 = 2,
setor_b1_x_0 = 10,
velocidade_barco1_0 = HIGHT,
rota_barco1_0 = 12]

```

Para verificar que não existe mais de 1 barco por setor, utilizamos o `bmc_always` para verificar a propriedade `setordif`, que utiliza a função `danger` entre todos os setores com a distancia 0.

```

[8]: def bmc_always(declare,init,trans,inv,K):
    for k in range(1,K+1):
        s = Solver()
        state=[declare(i) for i in range(k)]
        s.add(init(state[0]))
        for i in range(k-1):
            s.add(trans(state[i],state[i+1]))
        s.add(Not(inv(state[k-1])))
        if s.check()==sat:
            m=s.model()
            for i in range(k):
                print(m)
        print ("Property is valid up to traces of length "+str(K))

def setordif(s):
    return
    ↪Not(Or(danger(s, 'b1', 'b2', 0),danger(s, 'b1', 'b3', 0),danger(s, 'b2', 'b3', 0)))

bmc_always(declare,init,trans,setordif,3)

```

Property is valid up to traces of length 3

1.3 Conclusão:

Em estilo de conclusão, e como este é o último exercício realizado, para avaliação, no âmbito desta cadeira semestral, queríamos dizer que mais uma vez, este Trabalho 4 mais uma vez, fez com que pudessemos colocar em prática os conceitos teóricos adquiridos.

Este exercício foi “parecido” com o exercício 2, do Trabalho 3, mas obviamente necessitava de uma maior atenção e de mais alguns conhecimentos, que aprendemos entretanto.

Apesar de tudo isto, achamos que conseguimos concluir o exercício de uma forma muito positiva e cumprindo os objetivos pretendidos.

Esperemos que esteja do agrado do professor e queríamos ainda agradecer pela atenção ao longo do semestre.

[]: