

tp1_g20_Ex1

November 1, 2020

1 Lógica Computacional

Grupo 20

- Francisco Domingos Martins Oliveira, A82066
- José Luís Cerqueira Pires, A84552

1.1 Trabalho Prático 1

Vamos começar por importar as ferramentas necessárias em todo o Notebook.

```
[1]: from z3 import *  
import math
```

Vamos usar uma família $r_{p,s,d,h}$ de variáveis binárias (i.e., que assumem valores inteiros $\{0,1\}$), com a seguinte semântica

$r_{p,s,d,h} == 1$ se e só se o projeto p for alocado à sala s , no dia d , à hora h .

e também outra variável binária tal que:

$y_{p,c,s,d,h} == 1$ se e só se o colaborado c do projeto p for alocado à sala s , no dia d , à hora h .

```
[2]: horario = Solver()  
P = 2  
S = 3  
C = 3  
D = 3  
H = 4
```

Como input iremos usar dois dicionários, “Pro” que irá conter os projetos ao qual terá uma lista de colaboradores, o líder, o número de reuniões semanal. O “Dis” irá conter todos os colaboradores e uma lista dos dias e horas disponíveis. Além disso iremos usar dois arrays posteriormente preenchidos, um com todos os líderes e outro com todos os colaboradores em cada projeto para facilitar a escrita do código.

```
[3]: Pro={1:[[1,2],1,2] , 2:[[1,2],2,2]}  #{pojetos:[[colaboradores],lider,
    ↳n_reunioes semanais]}
Dis={1: [(1,1),(1,2),(1,3)],2: [(1,1),(1,2)]}#3: [(2,2),(2,3)]}#,3:
    ↳[(1,1),(1,3),(2,1)]}  #{colaboradores: [(dia disponivel,hora disponivel)]}
lideres = []  #lista com os lideres por ordem do numero de projeto -1 isto é.
    ↳lideres[0] é o lider do projeto 1
colaboradores = []  #lista com os colaboradores excetuando o por ordem do numero
    ↳de projeto -1 isto é. colabores[0] corresponde aos colaboradores do projeto 1
```

Escolhemos alocar duas matrizes , uma com os parâmetros “r”[P,S,D,H] e outra “y” [P,C,S,D,H].

Sendo $r[p,s,d,h]=1$ sse o projeto p for alocado à sala s, no dia d, à hora h e $y[p,c,s,d,h]=1$ sse o projeto p com o colaborador c for alocado à sala s, no dia d, à hora h

```
[4]: #Matrizes de alocação:
r = {}
for p in range(1,P+1):
    for s in range(1,S+1):
        for d in range(1,D+1):
            for h in range(1,H+1):
                r[p,s,d,h]=Int('r'+ '_' +str(p)+'_' +str(s)+'_' +str(d)+'_' +str(h))
                horario.add(r[p,s,d,h]<=1,r[p,s,d,h]>=0)

y = {}
for p in range(1,P+1):
    for c in range(1,C+1):
        for s in range(1,S+1):
            for d in range(1,D+1):
                for h in range(1,H+1):
                    ↳
                    ↳y[p,c,s,d,h]=Int('c'+ '_' +str(p)+'_' +str(c)+'_' +str(s)+'_' +str(d)+'_' +str(h))
                    ↳horario.add(y[p,c,s,d,h]<=1,y[p,c,s,d,h]>=0)

for p in range(1,P+1):
    lideres.append(Pro[p][1])

for p in range(1,P+1):
    l = []
    for c in Pro[p][0]:
        if(c!=Pro[p][1]):
            l.append(c)
    colaboradores.append(l)
```

Cada sala só pode ter no maximo uma reuniao a acontecer:

$$\forall_{1 \leq s \leq S} \cdot \quad \forall_{1 \leq d \leq D} \cdot \quad \forall_{1 \leq h \leq H} \cdot \quad \left(\sum_{1 \leq p \leq P} r_{p,s,d,h} \leq 1 \right)$$

```
[5]: #Cada sala só pode ter no maximo uma reuniao a acontecer:
for s in range(1,S+1):
    for d in range(1,D+1):
        for h in range(1,H+1):
            horario.add(Sum([r[p,s,d,h] for p in range (1,P+1)])<=1)
```

Em cada dia e cada hora cada projeto é alocado a uma ou nenhuma sala

$$\forall_{1 \leq d \leq D} \cdot \forall_{1 \leq h \leq H} \cdot \forall_{1 \leq p \leq P} \cdot \left(\sum_{1 \leq s \leq S} r_{p,s,d,h} \leq 1 \right)$$

```
[6]: #Em cada dia e cada hora o projeto é alocado a uma ou nenhuma sala: DONE
for d in range(1,D+1):
    for h in range(1,H+1):
        for p in range(1,P+1):
            horario.add(Sum([r[p,s,d,h] for s in range(1,S+1)])<=1)
```

Em cada dia e hora, cada colaborador é alocado a uma ou nenhuma sala Done

$$\forall_{1 \leq p \leq P} \cdot \forall_{1 \leq c \leq C} \cdot \forall_{(d,h) \in Dis[c]} \cdot \left(\sum_{1 \leq s \leq S} y_{p,c,s,d,h} \leq 1 \right)$$

```
[7]: #Em cada dia e hora, cada colaborador é alocado a uma ou nenhuma sala Done
for p in range (1,P+1):
    for d in range(1,D+1):
        for h in range(1,H+1):
            for c in range(1,C+1):
                horario.add(Sum([y[p,c,s,d,h] for s in range(1,S+1)])<=1)
```

Em cada dia e hora, cada líder é alocado a uma ou nenhuma sala Done

$$\forall_{1 \leq p \leq P} \cdot \forall_{l \in lideres[p-1]} \cdot \forall_{(d,h) \in Dis[l]} \cdot \left(\sum_{1 \leq s \leq S} y_{p,l,s,d,h} \leq 1 \right)$$

```
[8]: for p in range (1,P+1):
    l = lideres[p-1]
    for (d,h) in Dis[l]:
        horario.add(Sum([y[p,l,s,d,h] for s in range(1,S+1)])<=1)
```

A reuniao tem de ser em slots disponiveis do lider

$$\forall_{1 \leq p \leq P} \cdot \forall_{l \in lideres[p-1]} \cdot \left(\sum_{1 \leq s \leq S} \sum_{(d,h) \in Dis[l]} y_{p,l,s,d,h} == Pro[p][2] \right)$$

```
[9]: # a reuniao tem de ser em slots disponiveis do lider
for p in range(1,P+1):
    l = lideres[p-1]
```

```

    horario.add(Sum([r[p,s,d,h] for s in range(1,S+1) for (d,h) in Dis[1]])) ==
    →Pro[p][2])

```

Cada reuniao de projeto apenas pode ser 'N' vezes sendo 'N' o valor de input

$$\forall_{1 \leq p \leq P} \cdot \left(\sum_{1 \leq s \leq S} \sum_{1 \leq d \leq D} \sum_{1 \leq h \leq H} r_{p,s,d,h} == Pro[p][2] \right)$$

```

[10]: for p in range(1,P+1):
        horario.add(Sum([r[p,s,d,h] for s in range(1,S+1) for d in range(1,D+1) for
        →h in range(1,H+1)])) == Pro[p][2])

```

Os colaboradores podem ou não ir aos projetos que estão a acontecer no seu horario

$$\forall_{1 \leq p \leq P} \cdot \forall_{c \in colaboradores[p-1]} \cdot \left(\sum_{1 \leq s \leq S} \sum_{(d,h) \in Dis[c]} r_{p,s,d,h} \leq Pro[p][2] \right)$$

```

[11]: # os colaboradores podem ou não ir aos projetos que coincidem com o horario
        →deles
        for p in range(1,P+1):
            for c in colaboradores[p-1]:
                horario.add(Sum([r[p,s,d,h] for s in range(1,S+1) for (d,h) in Dis[c]]))
                →<= Pro[p][2])

```

O lider participa em todas as reunioes do seu projeto

$$\forall_{1 \leq s \leq S} \cdot \forall_{1 \leq d \leq D} \cdot \forall_{1 \leq h \leq H} \cdot \forall_{1 \leq p \leq P} \cdot \forall_{l \in lideres[p-1]} \cdot (r_{p,s,d,h} == y_{p,l,s,d,h})$$

```

[12]: for s in range(1,S+1):
        for d in range(1,D+1):
            for h in range(1,H+1):
                for p in range(1,P+1):
                    l = lideres[p-1]
                    horario.add(r[p,s,d,h] == y[p,l,s,d,h])

```

O colaborador pode ou nao participar em reunioes de projetos a qual faz parte

$$\forall_{1 \leq s \leq S} \cdot \forall_{1 \leq d \leq D} \cdot \forall_{1 \leq h \leq H} \cdot \forall_{1 \leq p \leq P} \cdot \forall_{c \in colaboradores[p-1]} \cdot (r_{p,s,d,h} \geq y_{p,c,s,d,h})$$

```

[13]: for s in range(1,S+1):
        for d in range(1,D+1):
            for h in range(1,H+1):
                for p in range(1,P+1):
                    for c in colaboradores[p-1]:
                        horario.add(r[p,s,d,h] >= y[p,c,s,d,h])

```

```
[14]: m=horario.check()
      if (m==sat):
          print("valido")
          hor = horario.model()

          for d in hor:
              if (int(hor[d].as_long())==1):
                  print("%s = %d" % (d.name(), hor[d].as_long()))
      else:
          print('invalido')
```

```
valido
c_2_1_2_1_2 = 1
r_2_1_1_1 = 1
c_2_2_2_1_2 = 1
c_1_1_3_1_3 = 1
c_1_1_1_1_2 = 1
c_2_2_1_1_1 = 1
r_2_2_1_2 = 1
r_1_1_1_2 = 1
r_1_3_1_3 = 1
```