

Número: pg53951

Nome: João Pedro Vilas Boas Braga

Curso: Mestrado em Engenharia Informática

Pretende-se colorir uma sequência de 9 azulejos justapostos com uma paleta de 4 cores: azul, verde, cinzento e preto. A coloração dos azulejos tem que ser feita de acordo com as seguintes regras:

1. Cada azulejo tem uma única cor.
2. Os azulejos das extremidades têm que ter a mesma cor.
3. Todas as cores têm que ser usadas.
4. O azulejo do meio só pode ser pintado de cinzento ou preto.
5. O penúltimo azulejo da sequência tem que ser preto, excepto se o último estiver pintado de preto.
6. Azulejos seguidos não podem ser pintados da mesma cor.
7. Caso o primeiro azulejo seja verde e o terceiro não seja preto, o segundo terá que ser cinzento ou preto.
8. Os azulejos na terceira e na sétima posição só podem ser pintados de azul ou de verde.

Pretende-se codificar este problema em lógica proposicional e usar o Z3 para responder a algumas questões.

```
!pip install z3-solver
```

```
Requirement already satisfied: z3-solver in /usr/local/lib/python3.10/dist-packages (4.12.2.0)
```

1. Codifique o problema num solver e comprove que o conjunto de fórmulas é consistente.

```
from z3 import *

cores = ["azul", "verde", "cinzento", "preto"]
azulejos = [1, 2, 3, 4, 5, 6, 7, 8, 9]
x = {}
for a in azulejos:
    x[a] = {}
    for c in cores:
        x[a][c] = Bool("%d,%s" % (a, c))

s = Solver()

# Cada azulejo tem uma única cor
for a in azulejos:
    s.add(Or([x[a][c] for c in cores]))
    s.add(Implies(x[a]["azul"], Not(Or(x[a]["verde"], x[a]["cinzento"], x[a]["preto"]))))
    s.add(Implies(x[a]["verde"], Not(Or(x[a]["cinzento"], x[a]["preto"]))))
    s.add(Implies(x[a]["cinzento"], Not(x[a]["preto"])))

# Os azulejos das extremidades têm que ter a mesma cor.
for c in cores:
    s.add(x[1][c] == x[9][c])

# Todas as cores têm que ser usadas.
for c in cores:
    s.add(Or([x[a][c] for a in azulejos])) # ??? if it works, it works

# O azulejo do meio só pode ser pintado de cinzento ou preto.
s.add(Or(x[5]["cinzento"], x[5]["preto"]))

# O penúltimo azulejo da sequência tem que ser preto, excepto se o último estiver pintado de preto.
s.add(Implies(Not(x[9]["preto"]), x[8]["preto"]))

# Azulejos seguidos não podem ser pintados da mesma cor.
for c in cores:
    for a in azulejos[:-1]:
        s.add(Implies(x[a][c], Not(x[a+1][c])))

# Caso o primeiro azulejo seja verde e o terceiro não seja preto, o segundo terá que ser cinzento ou preto.
s.add(Implies(And(x[1]["verde"], Not(x[3]["preto"])), Or(x[2]["cinzento"], x[2]["preto"])))

# Os azulejos na terceira e na sétima posição só podem ser pintados de azul ou de verde.
s.add(Or(x[3]["azul"], x[3]["verde"]))
s.add(Or(x[7]["azul"], x[7]["verde"]))

if s.check() == sat:
    m = s.model()
    for a in azulejos:
        for c in cores:
```

```

        if is_true(m[x[a][c]]):
            print("O azulejo %d é %s" % (a,c))
else:
    print("Não tem solução.")

```

```

➡ 0 azulejo 1 é azul
   0 azulejo 2 é cinzento
   0 azulejo 3 é verde
   0 azulejo 4 é azul
   0 azulejo 5 é cinzento
   0 azulejo 6 é verde
   0 azulejo 7 é azul
   0 azulejo 8 é preto
   0 azulejo 9 é azul

```

2. Com a ajuda do solver, pronuncie-se quanto à veracidade das seguintes afirmações:

a. Se o primeiro azulejo for preto, o penúltimo azulejo tem que ser azul.

```
s.push()
```

```

prop = "Se o primeiro azulejo for preto, o penúltimo azulejo tem que ser azul."
form = Implies(x[1]["preto"], x[8]["azul"])
s.add(Not(form))

```

```

if s.check() == sat :
    print("A afirmação: ", prop)
    print("é falsa.")
    m = s.model()
    print("Por exemplo:")
    for a in azulejos:
        for c in cores:
            if is_true(m[x[a][c]]):
                print("O azulejo %d é %s" % (a,c))
else:
    print("A afirmação: ", prop)
    print("é verdadeira.")

```

```
s.pop()
```

```

A afirmação: Se o primeiro azulejo for preto, o penúltimo azulejo tem que ser azul.
é falsa.
Por exemplo:
0 azulejo 1 é preto
0 azulejo 2 é azul
0 azulejo 3 é verde
0 azulejo 4 é azul
0 azulejo 5 é cinzento
0 azulejo 6 é verde
0 azulejo 7 é azul
0 azulejo 8 é verde
0 azulejo 9 é preto

```

b. Algum dos dois últimos azulejos tem que ser preto.

```
s.push()
```

```

prop = "Algum dos dois últimos azulejos tem que ser preto."
form = Or(x[8]["preto"], x[9]["preto"])
s.add(Not(form))

```

```

if s.check() == sat :
    print("A afirmação: ", prop)
    print("é falsa.")
    m = s.model()
    print("Por exemplo:")
    for a in azulejos:
        for c in cores:
            if is_true(m[x[a][c]]):
                print("O azulejo %d é %s" % (a,c))
else:
    print("A afirmação: ", prop)
    print("é verdadeira.")

```

```
s.pop()
```

```

A afirmação: Algum dos dois últimos azulejos tem que ser preto.
é verdadeira.

```

c. Algum dos três primeiros azulejos poderá ser azul.

```
s.push()

prop = "Algum dos três primeiros azulejos poderá ser azul."
form = Or(x[1]["azul"],x[2]["azul"],x[3]["azul"], And(Not(x[1]["azul"]),Not(x[2]["azul"]),Not(x[3]["azul"])))
s.add(Not(form))

if s.check() == sat :
    print("A afirmação: ", prop)
    print("é falsa.")
    m = s.model()
    print("Por exemplo:")
    for a in azulejos:
        for c in cores:
            if is_true(m[x[a][c]]):
                print("O azulejo %d é %s" % (a,c))
else:
    print("A afirmação: ", prop)
    print("é verdadeira.")

s.pop()

    A afirmação: Algum dos três primeiros azulejos poderá ser azul.
    é verdadeira.
```

3. Acrescente código ao programa de modo a saber todas as colorações que são possíveis fazer aos azulejos. Quantas são?

```
s.push()

i = 0

while s.check() == sat :
    i+=1
    m = s.model()
    f = []
    for a in azulejos:
        for c in cores:
            if is_true(m[x[a][c]]):
                print("O azulejo %d é %s" % (a,c))
                f.append(x[a][c])
    s.add(Not(And(f)))
    print()
print("Número de colorações possíveis: ", i)

s.pop()

    0 azulejo 1 é cinzento
    0 azulejo 2 é azul
    0 azulejo 3 é verde
    0 azulejo 4 é azul
    0 azulejo 5 é cinzento
    0 azulejo 6 é verde
    0 azulejo 7 é azul
    0 azulejo 8 é preto
    0 azulejo 9 é cinzento

    0 azulejo 1 é cinzento
    0 azulejo 2 é azul
    0 azulejo 3 é verde
    0 azulejo 4 é azul
    0 azulejo 5 é preto
    0 azulejo 6 é verde
    0 azulejo 7 é azul
    0 azulejo 8 é preto
    0 azulejo 9 é cinzento

    0 azulejo 1 é cinzento
    0 azulejo 2 é azul
    0 azulejo 3 é verde
    0 azulejo 4 é azul
    0 azulejo 5 é preto
    0 azulejo 6 é azul
    0 azulejo 7 é verde
    0 azulejo 8 é preto
    0 azulejo 9 é cinzento

    0 azulejo 1 é cinzento
    0 azulejo 2 é azul
    0 azulejo 3 é verde
    0 azulejo 4 é cinzento
    0 azulejo 5 é preto
```

```
0 azulejo 6 é azul
0 azulejo 7 é verde
0 azulejo 8 é preto
0 azulejo 9 é cinzento
```

```
0 azulejo 1 é cinzento
0 azulejo 2 é azul
0 azulejo 3 é verde
0 azulejo 4 é preto
0 azulejo 5 é cinzento
0 azulejo 6 é azul
0 azulejo 7 é verde
0 azulejo 8 é preto
0 azulejo 9 é cinzento
```

```
0 azulejo 1 é cinzento
0 azulejo 2 é azul
0 azulejo 3 é verde
0 azulejo 4 é preto
0 azulejo 5 é cinzento
0 azulejo 6 é preto
0 azulejo 7 é azul
0 azulejo 8 é preto
```