

Clique duas vezes (ou prima Enter) para editar.

Luís Filipe Cruz Sobral A89474 MIEI

O “Clube Desportivo de Ribeirão” tem as seguintes regras:

- Todos os sócios que usam bigode são casados.
- Cada sócio do clube que não é de Ribeirão tem que usar camisola amarela.
- Os sócios casados não podem assistir aos jogos ao Domingo.
- Um sócio vai aos jogos ao Domingo se e só se é de Ribeirão.
- Cada sócio usa bigode ou não usa camisola amarela.
- Todos os sócios de Ribeirão usam bigode.

1. Por forma a codificar este puzzle como problema SAT, defina um conjunto adequado de variáveis proposicionais, exprima as regras acima como fórmulas proposicionais, e converta essas fórmulas para CNF.

- a. Usar bigode
- b. Ser casado
- c. Ser de ribeirão
- d. Usar camisola amarela
- e. Ir aos jogos ao domingo

1. $a \rightarrow b \Leftrightarrow \neg a \vee b$
2. $\neg c \rightarrow d \Leftrightarrow c \vee d$
3. $b \rightarrow \neg e \Leftrightarrow \neg b \vee \neg e$
4. $e \Leftrightarrow c \Leftrightarrow (\neg e \vee c) \wedge (\neg c \vee e)$
5. $a \vee \neg d$
6. $c \rightarrow a \Leftrightarrow \neg c \vee a$

```
!pip install python-sat[pbplib,aiger]
```

```
from pysat.solvers import Minisat22
```

```
s = Minisat22()
```

```
s.add_clause([-1,2])
```

```
s.add_clause([3,4])
```

```
s.add_clause([-2,-5])
```

```
s.add_clause([-5,3])
```

```

s.add_clause([-3,5])
s.add_clause([1,-4])
s.add_clause([-3,1])

if s.solve():
    print("SAT")
    print(s.get_model())
else:
    print("UNSAT")

Requirement already satisfied: python-sat[aiger,pbllib] in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: pypbllib>=0.0.3 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: py-aiger-cnf>=2.0.0 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: py-aiger<7.0.0,>=6.0.0 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: funcy<2.0,>=1.12 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: bidict<0.22.0,>=0.21.0 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: attrs<21.0.0,>=20.0.0 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: toposort<2.0,>=1.5 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: parsimonious<0.9.0,>=0.8.1 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: sortedcontainers<3.0.0,>=2.3.0 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: pyrsistent<0.18.0,>=0.17.0 in /usr/local/lib/python3.7/dist-packages (
SAT
[1, 2, -3, 4, -5]

```

3. Use agora o SAT solver para o ajudar a responder às seguintes questões:

- (a) A afirmação “Quem usa bigode não pode ir ao jogo ao Domingo.” é correcta?
- (b) Pode um membro de camisola amarela ser casado?
- (c) A afirmação “Afinal o clube não pode ter sócios Ribeironenses.” é correcta?
- (d) Os sócios casados têm todos bigode?
- (e) A afirmação “Ao domingo nunca há sócios a assistir aos jogos.” é correcta?

Justifique as suas respostas.

```

# Quem usa bigode não pode ir ao jogo ao Domingo
# a -> -e <=> -a -> -e

```

```

from pysat.solvers import Minisat22

```

```

s = Minisat22()

```

```

s.add_clause([-1,2])
s.add_clause([3,4])
s.add_clause([-2,-5])
s.add_clause([-5,3])
s.add_clause([-3,5])
s.add_clause([1,-4])
s.add_clause([-3,1])
s.add_clause([-1,-5])

```

```

if s.solve():
    print("SAT")

```

```

    print(s.get_model())
else:
    print("UNSAT")

    SAT
    [1, 2, -3, 4, -5]

```

```

# Pode um membro de camisola amarela ser casado
#  $d \wedge b$ 

```

```
from pysat.solvers import Minisat22
```

```
s = Minisat22()
```

```

s.add_clause([-1,2])
s.add_clause([3,4])
s.add_clause([-2,-5])
s.add_clause([-5,3])
s.add_clause([-3,5])
s.add_clause([1,-4])
s.add_clause([-3,1])
s.add_clause([4])
s.add_clause([2])

```

```

if s.solve():
    print("SAT")
    print(s.get_model())
else:
    print("UNSAT")

```

```

↳ SAT
   [1, 2, -3, 4, -5]

```

```

# Afinal o clube não pode ter sócios Ribeironenses.
#  $\neg c$ 

```

```
from pysat.solvers import Minisat22
```

```
s = Minisat22()
```

```

s.add_clause([-1,2])
s.add_clause([3,4])
s.add_clause([-2,-5])
s.add_clause([-5,3])
s.add_clause([-3,5])
s.add_clause([1,-4])
s.add_clause([-3,1])
s.add_clause([-3])

```

```

if s.solve():
    print("SAT")
    print(s.get_model())
else:
    print("UNSAT")

```

SAT

[1, 2, -3, 4, -5]

Os sócios casados têm todos bigode?

$b \rightarrow a \Leftrightarrow \neg b \vee a$

```
from pysat.solvers import Minisat22
```

```
s = Minisat22()
```

```
s.add_clause([-1,2])
```

```
s.add_clause([3,4])
```

```
s.add_clause([-2,-5])
```

```
s.add_clause([-5,3])
```

```
s.add_clause([-3,5])
```

```
s.add_clause([1,-4])
```

```
s.add_clause([-3,1])
```

```
s.add_clause([-2,1])
```

```
if s.solve():
```

```
    print("SAT")
```

```
    print(s.get_model())
```

```
else:
```

```
    print("UNSAT")
```

SAT

[1, 2, -3, 4, -5]

DE forma a testar a veracidade da afirmação iremos testar a negação da mesma

$\neg (\neg b \vee a) \Leftrightarrow b \wedge \neg a$

```
from pysat.solvers import Minisat22
```

```
s = Minisat22()
```

```
s.add_clause([-1,2])
```

```
s.add_clause([3,4])
```

```
s.add_clause([-2,-5])
```

```
s.add_clause([-5,3])
```

```
s.add_clause([-3,5])
```

```
s.add_clause([1,-4])
```

```
s.add_clause([-3,1])
```

```
s.add_clause([2])
```

```
s.add_clause([-1])
```

```
if s.solve():
```

```
    print("SAT")
```

```
    print(s.get_model())
```

```
else:
```

```
    print("UNSAT")
```

UNSAT

R: Como não existe um modelo que satisfaça a negação da afirmação, então podemos concluir que não existe um sócio casado que tenho bigode.

```
# Ao domingo nunca há sócios a assistir aos jogos.
# -e

from pysat.solvers import Minisat22

s = Minisat22()

s.add_clause([-1,2])
s.add_clause([3,4])
s.add_clause([-2,-5])
s.add_clause([-5,3])
s.add_clause([-3,5])
s.add_clause([1,-4])
s.add_clause([-3,1])
s.add_clause([-5])

if s.solve():
    print("SAT")
    print(s.get_model())
else:
    print("UNSAT")

    SAT
    [1, 2, -3, 4, -5]

# De forma a testarmos a veracidade da afirmação iremos testar a negação da mesma
#  $\neg(\neg e) \Leftrightarrow e$ 

from pysat.solvers import Minisat22

s = Minisat22()

s.add_clause([-1,2])
s.add_clause([3,4])
s.add_clause([-2,-5])
s.add_clause([-5,3])
s.add_clause([-3,5])
s.add_clause([1,-4])
s.add_clause([-3,1])
s.add_clause([5])

if s.solve():
    print("SAT")
    print(s.get_model())
else:
    print("UNSAT")
```

R: Podemos concluir que ao domingo há sócios a assistir aos jogos.

