

pg47398 - Leonardo de Freitas Marreiros

MEI

## ▼ Clube Desportivo de Ribeirão

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.**

Seja:

1. A - Sócio usa bigode
2. B - Sócio é de Ribeirão
3. C - Sócio é casado
4. D - Sócio usa camisola amarela
5. E - Sócio assiste aos jogos de domingo

Todos os sócios que usam bigode são casados.  $A \rightarrow C \equiv \neg A \vee C$

Cada sócio do clube que não é de Ribeirão tem que usar camisola amarela.  $\neg B \rightarrow D \equiv B \vee D$

Os sócios casados não podem assistir aos jogos ao Domingo.  $C \rightarrow \neg E \equiv \neg C \vee \neg E$

Um sócio vai aos jogos ao Domingo se e só se é de Ribeirão.  $E \leftrightarrow B \equiv (\neg E \vee B) \wedge (\neg B \vee E)$

Cada sócio usa bigode ou não usa camiseta amarela.  $A \vee \neg D$

Todos os sócios de Ribeirão usam bigode.  $B \rightarrow A \equiv \neg B \vee A$

## 2. Codifique o problema num SAT solver e comprove que o conjunto de regras é consistente.

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

```
Collecting python-sat[aiger,pbllib]
  Downloading python_sat-0.1.7.dev11-cp37-cp37m-manylinux2010_x86_64.whl (1.8 MB)
    |████████████████████████████████████████| 1.8 MB 4.3 MB/s
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from python-sat[aiger,pbllib]) (1.15.0)
Collecting py-aiger-cnf>=2.0.0
  Downloading py_aiger_cnf-5.0.2-py3-none-any.whl (5.2 kB)
Collecting pypbllib>=0.0.3
  Downloading pypbllib-0.0.4-cp37-cp37m-manylinux2014_x86_64.whl (3.4 MB)
    |████████████████████████████████████████| 3.4 MB 21.1 MB/s
Collecting py-aiger<7.0.0,>=6.0.0
  Downloading py_aiger-6.1.14-py3-none-any.whl (18 kB)
Collecting funcy<2.0,>=1.12
  Downloading funcy-1.16-py2.py3-none-any.whl (32 kB)
Collecting bidict<0.22.0,>=0.21.0
  Downloading bidict-0.21.3-py3-none-any.whl (36 kB)
Collecting toposort<2.0,>=1.5
  Downloading toposort-1.7-py2.py3-none-any.whl (9.0 kB)
Collecting parsimonious<0.9.0,>=0.8.1
  Downloading parsimonious-0.8.1.tar.gz (45 kB)
    |████████████████████████████████████████| 45 kB 2.5 MB/s
Requirement already satisfied: sortedcontainers<3.0.0,>=2.3.0 in /usr/local/lib/python3.7/dist-packages (from py-aiger<7.0.0,>=6.0.0)
Collecting attrs<21.0.0,>=20.0.0
  Downloading attrs-20.3.0-py2.py3-none-any.whl (49 kB)
    |████████████████████████████████████████| 49 kB 5.2 MB/s
Collecting pyrsistent<0.18.0,>=0.17.0
  Downloading pyrsistent-0.17.3.tar.gz (106 kB)
    |████████████████████████████████████████| 106 kB 54.9 MB/s
Building wheels for collected packages: parsimonious, pyrsistent
  Building wheel for parsimonious (setup.py) ... done
  Created wheel for parsimonious: filename=parsimonious-0.8.1-py3-none-any.whl size=42724 sha256=3bd4100a5ffe534cb50cf5bde35461
  Stored in directory: /root/.cache/pip/wheels/88/5d/ba/f27d8af07306b65ee44f9d3f9cadea1db749a421a6db8a99bf
  Building wheel for pyrsistent (setup.py) ... done
  Created wheel for pyrsistent: filename=pyrsistent-0.17.3-cp37-cp37m-linux_x86_64.whl size=98074 sha256=8a6a5ab0723ee20a0c7fd3
```

```

Stored in directory: /root/.cache/pip/wheels/a5/52/bf/71258a1d7b3c8cbe1ee53f9314c6f65f20385481eaae573cc5
Successfully built parsimonious pyrsistent
Installing collected packages: toposort, pyrsistent, parsimonious, funcy, bidict, attrs, py-aiger, python-sat, pypblib, py-aige
Attempting uninstall: pyrsistent
  Found existing installation: pyrsistent 0.18.0
  Uninstalling pyrsistent-0.18.0:
    Successfully uninstalled pyrsistent-0.18.0
Attempting uninstall: attrs
  Found existing installation: attrs 21.2.0
  Uninstalling attrs-21.2.0:
    Successfully uninstalled attrs-21.2.0
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is th
datascience 0.10.6 requires folium==0.2.1, but you have folium 0.8.3 which is incompatible.
Successfully installed attrs-20.3.0 bidict-0.21.3 funcy-1.16 parsimonious-0.8.1 py-aiger-6.1.14 py-aiger-cnf-5.0.2 pypblib-0.0.

```

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

```
s = Minisat22()
```

```
rules = ['A', 'B', 'C', 'D', 'E']
```

```
x = {}
```

```
c = 1
```

```
for r in rules:
```

```
    x[r] = c
```

```
    c += 1
```

```
s.add_clause([-x['A'], x['C']])
```

```
s.add_clause([x['B'], x['D']])
```

```
s.add_clause([-x['C'], -x['E']])
```

```
s.add_clause([-x['E'], x['B']])
```

```
s.add_clause([-x['B'], x['E']])
```

```
s.add_clause([x['A'], -x['D']])
```

```
s.add_clause([-x['B'], x['A']])
```

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

```
    m = s.get_model()
```

```
    print(m)
```

```

for w in rules:
    if m[x[w]-1]>0:
        if m[x[w]-1]==1:
            print("Os sócios usam bigode - %s." % w)
        if m[x[w]-1]==2:
            print("Os sócios são de Ribeirão - %s." % w)
        if m[x[w]-1]==3:
            print("Os sócios são casados - %s." % w)
        if m[x[w]-1]==4:
            print("Os sócios usam camisola amarela - %s." % w)
        if m[x[w]-1]==5:
            print("Os sócios assistem aos jogos de domingo - %s." % w)
    else:
        print("UNSAT")
s.delete()

[1, -2, 3, 4, -5]
Os sócios usam bigode - A.
Os sócios são casados - C.
Os sócios usam camisola amarela - D.

```

### 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?

Quem usa bigode não pode ir ao jogo ao Domingo.  $A \rightarrow \neg E \equiv \neg A \vee \neg E$

Se a negação da afirmação for UNSAT então a afirmação é verdadeira.  $A \wedge E$

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

```
s = Minisat22()
```

```
rules = ['A', 'B', 'C', 'D', 'E']
```

```
x = {}
```

```

c = 1
for r in rules:
    x[r] = c
    c += 1

s.add_clause([-x['A'], x['C']])
s.add_clause([x['B'], x['D']])
s.add_clause([-x['C'], -x['E']])
s.add_clause([-x['E'], x['B']])
s.add_clause([-x['B'], x['E']])
s.add_clause([x['A'], -x['D']])
s.add_clause([-x['B'], x['A']])
s.add_clause([x['A']])
s.add_clause([x['E']])

if s.solve():
    m = s.get_model()
    print(m)
    for w in rules:
        if m[x[w]-1]>0:
            if m[x[w]-1]==1:
                print("Os sócios usam bigode - %s." % w)
            if m[x[w]-1]==2:
                print("Os sócios são de Ribeirão - %s." % w)
            if m[x[w]-1]==3:
                print("Os sócios são casados - %s." % w)
            if m[x[w]-1]==4:
                print("Os sócios usam camisola amarela - %s." % w)
            if m[x[w]-1]==5:
                print("Os sócios assistem aos jogos de domingo - %s." % w)
    else:
        print("UNSAT")
s.delete()

UNSAT

```

A afirmação é verdadeira.

**b)** Pode um membro de camisola amarela ser casado?

Pode um membro de camisola amarela ser casado? **D**  $\wedge$  **C**

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

```
s = Minisat22()
```

```
rules = ['A','B','C','D','E']
```

```
x = {}
```

```
c = 1
```

```
for r in rules:
```

```
    x[r] = c
```

```
    c += 1
```

```
s.add_clause([-x['A'], x['C']])
```

```
s.add_clause([x['B'], x['D']])
```

```
s.add_clause([-x['C'], -x['E']])
```

```
s.add_clause([-x['E'], x['B']])
```

```
s.add_clause([-x['B'], x['E']])
```

```
s.add_clause([x['A'], -x['D']])
```

```
s.add_clause([-x['B'], x['A']])
```

```
s.add_clause([x['D']])
```

```
s.add_clause([x['C']])
```

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

```
    m = s.get_model()
```

```
    print(m)
```

```
    for w in rules:
```

```
        if m[x[w]-1]>0:
```

```
            if m[x[w]-1]==1:
```

```
                print("Os sócios usam bigode - %s." % w)
```

```

    if m[x[w]-1]==2:
        print("Os sócios são de Ribeirão - %s." % w)
    if m[x[w]-1]==3:
        print("Os sócios são casados - %s." % w)
    if m[x[w]-1]==4:
        print("Os sócios usam camisola amarela - %s." % w)
    if m[x[w]-1]==5:
        print("Os sócios assistem aos jogos de domingo - %s." % w)
else:
    print("UNSAT")
s.delete()

[1, -2, 3, 4, -5]
Os sócios usam bigode - A.
Os sócios são casados - C.
Os sócios usam camisola amarela - D.

```

Sim.

c) A afirmação "Afinal o clube não pode ter sócios Ribeironenses" é correcta?

Afinal o clube não pode ter sócios Ribeironenses. **¬B**

Se a negação da afirmação for UNSAT então a afirmação é verdadeira. **B**

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

```
s = Minisat22()
```

```
rules = ['A', 'B', 'C', 'D', 'E']
```

```
x = {}
```

```
c = 1
```

```
for r in rules:
```

```
    x[r] = c
```

```
    c += 1
```

```

s.add_clause([-x['A'], x['C']])
s.add_clause([x['B'], x['D']])
s.add_clause([-x['C'], -x['E']])
s.add_clause([-x['E'], x['B']])
s.add_clause([-x['B'], x['E']])
s.add_clause([x['A'], -x['D']])
s.add_clause([-x['B'], x['A']])
s.add_clause([x['B']])

if s.solve():
    m = s.get_model()
    print(m)
    for w in rules:
        if m[x[w]-1]>0:
            if m[x[w]-1]==1:
                print("Os sócios usam bigode - %s." % w)
            if m[x[w]-1]==2:
                print("Os sócios são de Ribeirão - %s." % w)
            if m[x[w]-1]==3:
                print("Os sócios são casados - %s." % w)
            if m[x[w]-1]==4:
                print("Os sócios usam camiseta amarela - %s." % w)
            if m[x[w]-1]==5:
                print("Os sócios assistem aos jogos de domingo - %s." % w)
    else:
        print("UNSAT")
s.delete()

UNSAT

```

A afirmação é verdadeira.

**d)** Os sócios casados têm todos bigode?



Os sócios casados têm todos bigode?  $C \rightarrow A \equiv \neg C \vee A$

```

from pysat.solvers import Minisat22

s = Minisat22()

rules = ['A', 'B', 'C', 'D', 'E']
x = {}
c = 1
for r in rules:
    x[r] = c
    c += 1

s.add_clause([-x['A'], x['C']])
s.add_clause([x['B'], x['D']])
s.add_clause([-x['C'], -x['E']])
s.add_clause([-x['E'], x['B']])
s.add_clause([-x['B'], x['E']])
s.add_clause([x['A'], -x['D']])
s.add_clause([-x['B'], x['A']])
s.add_clause([-x['C'], x['A']])

if s.solve():
    m = s.get_model()
    print(m)
    for w in rules:
        if m[x[w]-1]>0:
            if m[x[w]-1]==1:
                print("Os sócios usam bigode - %s." % w)
            if m[x[w]-1]==2:
                print("Os sócios são de Ribeirão - %s." % w)
            if m[x[w]-1]==3:
                print("Os sócios são casados - %s." % w)
            if m[x[w]-1]==4:

```

```

        print("Os sócios usam camisola amarela - %s." % w)
    if m[x[w]-1]==5:
        print("Os sócios assistem aos jogos de domingo - %s." % w)
else:
    print("UNSAT")
s.delete()

[1, -2, 3, 4, -5]
Os sócios usam bigode - A.
Os sócios são casados - C.
Os sócios usam camisola amarela - D.

```

Sim.

**e)** A afirmação "Ao domingo nunca há sócios a assistir aos jogos" é correcta?

Ao domingo nunca há sócios a assistir aos jogos. **¬E**

Se a negação da afirmação for UNSAT então a afirmação é verdadeira. **E**

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

```
s = Minisat22()
```

```
rules = ['A', 'B', 'C', 'D', 'E']
```

```
x = {}
```

```
c = 1
```

```
for r in rules:
```

```
    x[r] = c
```

```
    c += 1
```

```
s.add_clause([-x['A'], x['C']])
```

```
s.add_clause([x['B'], x['D']])
```

```
s.add_clause([-x['C'], -x['E']])
```

```
s.add_clause([-x['E'], x['B']])
```

```

s.add_clause([-x['B'], x['E']])
s.add_clause([x['A'], -x['D']])
s.add_clause([-x['B'], x['A']])
s.add_clause([x['E']])

if s.solve():
    m = s.get_model()
    print(m)
    for w in rules:
        if m[x[w]-1]>0:
            if m[x[w]-1]==1:
                print("Os sócios usam bigode - %s." % w)
            if m[x[w]-1]==2:
                print("Os sócios são de Ribeirão - %s." % w)
            if m[x[w]-1]==3:
                print("Os sócios são casados - %s." % w)
            if m[x[w]-1]==4:
                print("Os sócios usam camisola amarela - %s." % w)
            if m[x[w]-1]==5:
                print("Os sócios assistem aos jogos de domingo - %s." % w)
    else:
        print("UNSAT")
s.delete()

```

UNSAT

Sim.

