

**A74806 - João Luís Pereira Amorim - MIEI**

```
!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 6.7 MB/s
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from py
Collecting pypbllib>=0.0.3
  Downloading pypbllib-0.0.4-cp37-cp37m-manylinux2014_x86_64.whl (3.4 MB)
    |████████████████████████████████████████| 3.4 MB 67.6 MB/s
Collecting py-aiger-cnf>=2.0.0
  Downloading py_aiger_cnf-5.0.2-py3-none-any.whl (5.2 kB)
Collecting bidict<0.22.0,>=0.21.0
  Downloading bidict-0.21.3-py3-none-any.whl (36 kB)
Collecting funcy<2.0,>=1.12
  Downloading funcy-1.16-py2.py3-none-any.whl (32 kB)
Collecting py-aiger<7.0.0,>=6.0.0
  Downloading py_aiger-6.1.14-py3-none-any.whl (18 kB)
Collecting pyrsistent<0.18.0,>=0.17.0
  Downloading pyrsistent-0.17.3.tar.gz (106 kB)
    |████████████████████████████████████████| 106 kB 53.8 MB/s
Collecting parsimonious<0.9.0,>=0.8.1
  Downloading parsimonious-0.8.1.tar.gz (45 kB)
    |████████████████████████████████████████| 45 kB 2.7 MB/s
Requirement already satisfied: sortedcontainers<3.0.0,>=2.3.0 in /usr/local/lib/pytho
Collecting attrs<21.0.0,>=20.0.0
  Downloading attrs-20.3.0-py2.py3-none-any.whl (49 kB)
    |████████████████████████████████████████| 49 kB 5.7 MB/s
Collecting toposort<2.0,>=1.5
  Downloading toposort-1.7-py2.py3-none-any.whl (9.0 kB)
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=4
  Stored in directory: /root/.cache/pip/wheels/88/5d/ba/f27d8af07306b65ee44f9d3f9cad6
  Building wheel for pyrsistent (setup.py) ... done
  Created wheel for pyrsistent: filename=pyrsistent-0.17.3-cp37-cp37m-linux_x86_64.whl
  Stored in directory: /root/.cache/pip/wheels/a5/52/bf/71258a1d7b3c8cbe1ee53f9314c6f
Successfully built parsimonious pyrsistent
Installing collected packages: toposort, pyrsistent, parsimonious, funcy, bidict, att
  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 package
datascience 0.10.6 requires folium==0.2.1, but you have folium 0.8.3 which is incom
Successfully installed attrs-20.3.0 bidict-0.21.3 funcy-1.16 parsimonious-0.8.1 py-a
```

1. De acordo com as regras fornecidas sobre o "Clube Desportivo de Ribeirão", é possível verificar as seguintes variáveis proposicionais:

B: Sócio que usa Bigode.

C: Sócio que é Casado.

R: Sócio que é de Ribeirão.

A: Sócio que usa camisola Amarela.

D: Sócio que assiste aos jogos ao Domingo.

Caso quisessemos utilizar o formato standard para SAT Solvers DIMACS CNF, B seria 1, C seria 2, R seria 3, A seria 4 e D seria 5. Para este TPC, serão usadas as letras.

Quanto às fórmulas proposicionais e as suas correspondentes formas CNF, foi obtido o seguinte (Em caso de ser necessária uma conversão para CNF, o resultado desta encontra-se sempre do lado direito):

- Todos os sócios que usam bigode são casados.

$$(B \rightarrow C) \equiv (\neg B \vee C)$$

- Cada sócio do clube que não é de Ribeirão tem que usar camisola amarela.

$$(\neg R \rightarrow A) \equiv (R \vee A)$$

- Os sócios casados não podem assistir aos jogos ao Domingo

$$(C \rightarrow \neg D) \equiv (\neg C \vee \neg D)$$

- Um sócio vai aos jogos ao Domingo se e só se é de Ribeirão.

$$(D \leftrightarrow R) \equiv ((D \rightarrow R) \wedge (R \rightarrow D)) \equiv (\neg D \vee R) \wedge (\neg R \vee D)$$

- Cada sócio usa bigode ou não usa camisola amarela.

$$(B \vee \neg A)$$

- Todos os sócios de Ribeirão usam bigode.

$$(R \rightarrow B) \equiv (\neg R \vee B)$$

Em resumo, obteram-se as seguintes fórmulas em CNF:

```
(¬B ∨ C)
(R ∨ A)
(¬C ∨ ¬D)
(¬D ∨ R) ∧ (¬R ∨ D)
(B ∨ ¬A)
(¬R ∨ B)
```

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

```
from pysat.solvers import Minisat22

s = Minisat22()

restritions = ['Bigode', 'Casado', 'Ribeirao', 'Amarela', 'Domingo']
x = {}
c = 1
for r in restritions:
    x[r] = c
    c += 1

s.add_clause([-x['Bigode'], x['Casado']])
s.add_clause([x['Ribeirao'], x['Amarela']])
s.add_clause([-x['Casado'], -x['Domingo']])
s.add_clause([-x['Domingo'], x['Ribeirao']])
s.add_clause([-x['Ribeirao'], x['Domingo']])
s.add_clause([x['Bigode'], -x['Amarela']])
s.add_clause([-x['Ribeirao'], x['Bigode']])

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

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

Tendo em conta que foi obtida uma solução que satisfaz o conjunto de regras apresentadas, podemos concluir que o mesmo é consistente. Concluimos que os sócios do Clube Desportivo de Ribeirão têm Bigode, são Casados, não são de Ribeirão, usam Camisola Amarela e não assistem aos jogos ao Domingo.

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

Nesta secção, para afirmações em que é necessário que estas sejam sempre verdade, iremos utilizar a negação da fórmula proposicional, em que se o resultado do solver for UNSAT, nos indicará que a afirmação será correta, enquanto que nas outras, as cláusulas serão adicionadas ao solver normalmente, cujo resultado esperado para que estas sejam corretas será SAT.

a. A afirmação "Quem usa bigode não pode ir ao jogo ao Domingo" é correcta?

Fórmula proposicional:

$$(B \rightarrow \neg D) \equiv (\neg B \vee \neg D)$$

A negação transforma em:

$$\neg(\neg B \vee \neg D) \equiv B \wedge D$$

```
s.add_clause([x['Bigode']])
s.add_clause([x['Domingo']])
```

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

UNSAT

Esta afirmação é de facto correta, uma vez que depois de adicionadas as cláusulas correspondentes à negação da fórmula proposicional, foi obtido, tal como dito inicialmente, UNSAT.

b. Pode um membro de camisola amarela ser casado?

Fórmula proposicional:

$$A \wedge C$$

A

C

```
s.add_clause([x['Amarela']])
s.add_clause([x['Casado']])
```

```
if s.solve():
    m = s.get_model()
    print(m)
```

```

    print("SAT")
else:
    print("UNSAT")

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

```

Um membro de camisola amarela pode ser de facto casado, uma vez que após adicionadas as cláusulas correspondentes à fórmula proposicional " $A \wedge C$ ", continuam a existir soluções que tornam o conjunto de regras satisfazíveis.

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

Fórmula proposicional:

$\neg R$

A negação transforma em:

$\neg(\neg R) \equiv R$

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

```

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

    UNSAT

```

Esta afirmação é de facto correcta, uma vez que depois de adicionada a cláusula correspondente à negação da fórmula proposicional, foi obtido, tal como dito inicialmente, UNSAT.

d. Os sócios casados têm todos bigode?

Fórmula proposicional:

$C \rightarrow B \equiv \neg C \vee B$

A negação transforma em:

$\neg(\neg C \vee B) \equiv C \wedge \neg B$

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

```
s.add_clause([-x['Bigode']])
```

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

UNSAT

Esta afirmação é de facto correta, uma vez que depois de adicionada a clausula correspondente à negação da fórmula proposicional, foi obtido, tal como dito inicialmente, UNSAT.

e. A afirmação “Ao domingo nunca há sócios a assistir aos jogos.” é correcta?

Fórmula proposicional:

$\neg D$

A negação transforma em:

$\neg(\neg D) \equiv D$

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

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

UNSAT

Esta afirmação é de facto correta, uma vez que depois de adicionada a clausula correspondente à negação da fórmula proposicional, foi obtido, tal como dito inicialmente, UNSAT.

