pg46988, Ana Rita Abreu Peixoto, Mestrado em Engenharia Informática

Exercício 1:

As variáveis proposicionais consideradas na resolução do problema são as seguintes:

- p1: Sócio com bigode
- p2: Sócio casado
- p3: Sócio de Ribeirão
- p4: Sócio utiliza camisola amarela
- p5: Sócio assiste aos jogos de domingo

Estas variáveis permitem codificar as seguintes regras como fórmulas proposicionais e respetiva fórmula CNF, do seguinte modo:

- **"Todos os sócios que usam bigode são casados."**

  Fórmula proposicional:

  `p1 ⇒ p2`

  Fórmula CNF:

  `¬p1 V p2`

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

  Fórmula proposicional:

  `¬p3 ⇒ p4`

  Fórmula CNF:

  `p3 V p4`

- **"Os sócios casados não podem assistir aos jogos ao Domingo."**

  Fórmula proposicional:

  `p2 ⇒ ¬p5`

  Fórmula CNF:

  `¬p2 V ¬p5`

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

  Fórmula proposicional:

  `p5 ⇔ p3`

Fórmula CNF:

`(¬p3 V p5) ∧ (¬p5 V p3)`

- **"Cada sócio usa bigode ou não usa camisola amarela."**

  Fórmula proposicional e *CNF*:

  `p1 V ¬p4`

- **"Todos os sócios de Ribeirão usam bigode."**

  Fórmula proposicional:

  `p3 ⇒ p1`

  Fórmula CNF:

  `¬p3 V p1`

Deste modo, a fórmula F no formato CNF que traduz o problema é:

```
F = (¬p1 V p2) ∧ (p3 V p4) ∧ (¬p2 V ¬p5) ∧ (¬p3 V p5) ∧ (¬p5 V p3) ∧ (p1 V ¬p4) ∧
(¬p3 V p1)
```

Exercício 2:

Um conjunto de regras é consistente se existe pelo menos uma possibilidade em que todas as regras são verdadeiras, ou seja, é **consistente** se **não é uma contradição**.

Para provar que o conjunto das regras acima definidas é **consistente**, é necessário recorrer a um SAT Solver e converter a fórmula F para o formato DIMACS CNF, tal como se apresenta abaixo.

Foi criado um ficheiro ex2.cnf que irá ser passado como input ao SAT Solver. Este ficheiro pode ser observado de seguida e denota a fórmula F, referida anteriormente.

```
p cnf 5 7
-1 2 0
3 4 0
-2 -5 0
-3 5 0
-5 3 0
1 -4 0
-3 1 0
```

Após executar o comando "minisat ex2.cnf OUT" é possível observar o output gerado pelo solver no ficheiro OUT, que se apresenta de seguida:

```
SAT
1 2 -3 4 -5 0
```

A resposta do solver minisat é SATISFIABLE, o que significa que o conjunto das regras que definem o problema é **consistente**.

Além disso, também apresenta uma resposta possível ao problema em questão, atribuindo valores de verdade às variáveis que o definem. Deste modo, p1, p2 e p4 possuem valoração **Verdadeira** e as variáveis p3 e p5 são **Falsas**.

Exercício 3:

Com o auxílio de um SAT Solver é possível responder a algumas questões pertinentes no âmbito do problema, tais como:

```
Seja Γ = { (¬p1 V p2) , (p3 V p4) , (¬p2 V ¬p5) , (¬p3 V p5) , (¬p5 V p3) , (p1 V ¬p4) ,
(¬p3 V p1) }
```

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

   Esta afirmação pode ser codificada através de variáveis proposicionais da seguinte forma:

   `p1 ⇒ ¬p5`

   E é representada em CNF por:

   `¬p1 V ¬p5`

   Para obter resposta a esta questão é necessário que: `Γ ⊨ (¬p1 V ¬p5)` , ou seja, que `F ⇒ (¬p1 V ¬p5)` seja válido.

```
F ⇒ (¬p1 V ¬p5) é equivalente a ¬F V (¬p1 V ¬p5) e para provar a sua validez é necessário provar
que ¬(¬F V (¬p1 V ¬p5)) é UNSAT.

Assim, queremos provar que F ∧ ¬(¬p1 V ¬p5) é UNSAT, ou seja, que F ∧ (p1 ∧ p5) é UNSAT.
```

Para isso podemos recorrer ao SAT Solver minisat, escrevendo o seguinte ficheiro DIMACS CNF:

```
p cnf 5 9
-1 2 0
3 4 0
-2 -5 0
-3 5 0
-5 3 0
1 -4 0
-3 1 0
1 0 c primeira restrição adicionada
5 0 c segunda restrição adicionada
```

O resultado obtido pelo solver minisat é UNSAT, logo a afirmação é **correta**.

2. **Pode um membro de camisola amarela ser casado?**

   No formato CNF, esta afirmação representa-se do seguinte modo:

   `p4 ∧ p2`

   Para obter resposta a esta questão é necessário que: `Γ ⊨ (p4 ∧ p2)`, ou seja, que `F ⇒ (p4 ∧ p2)` seja válido.

```
F ⇒ (p4 ∧ p2) é equivalente a ¬F V (p4 ∧ p2) e para provar a sua validez é necessário provar
que ¬(¬F V (p4 ∧ p2)) é UNSAT.

Assim, queremos provar que F ∧ ¬(p4 ∧ p2) é UNSAT, ou seja, que F ∧ (¬p4 V ¬p2) é UNSAT.
```

Para isso podemos recorrer ao SAT Solver minisat, escrevendo o seguinte ficheiro DIMACS CNF:

```
p cnf 5 8
-1 2 0
3 4 0
-2 -5 0
-3 5 0
-5 3 0
1 -4 0
-3 1 0
-4 -2 0 c restrição adicionada
```

O resultado obtido pelo solver minisat é UNSAT, logo a afirmação é **correta**.

3. **A afirmação “Afinal o clube não pode ter sócios Ribeironenses.” é correcta?**

   No formato CNF, esta afirmação representa-se do seguinte modo:

   `¬p3`

   Para obter resposta a esta questão é necessário que: `Γ ⊨ ¬p3`, ou seja, que `F ⇒ ¬p3` seja válido.

```
F ⇒ ¬p3 é equivalente a ¬F V ¬p3 e para provar a sua validez é necessário provar
que ¬(¬F V ¬p3) é UNSAT.

Assim, queremos provar que F ∧ ¬(¬p3) é UNSAT, ou seja, que F ∧ p3 é UNSAT.
```

Para isso podemos recorrer ao SAT Solver minisat, escrevendo o seguinte ficheiro DIMACS CNF:

```
p cnf 5 8
-1 2 0
3 4 0
-2 -5 0
-3 5 0
-5 3 0
1 -4 0
-3 1 0
3 0 c restrição adicionada
```

O resultado obtido pelo solver minisat é UNSAT, logo a afirmação é **correta**.

4. **Os sócios casados têm todos bigode?**

   No formato CNF, esta afirmação representa-se do seguinte modo:

   `p2 ⇒ p1`

   Para obter resposta a esta questão é necessário que: `Γ ⊨ (p2 ⇒ p1)`, ou seja, que `F ⇒ (p2 ⇒ p1)` seja válido.

```
F ⇒ (p2 ⇒ p1) é equivalente a ¬F V (p2 ⇒ p1) e para provar a sua validez é necessário provar
que ¬(¬F V (p2 ⇒ p1)) é UNSAT.

Assim, queremos provar que F ∧ ¬(p2 ⇒ p1) é UNSAT, ou seja, que F ∧ (p2 ∧ ¬p1) é UNSAT.
```

Para isso podemos recorrer ao SAT Solver minisat, escrevendo o seguinte ficheiro DIMACS CNF:

```
p cnf 5 8
-1 2 0
3 4 0
-2 -5 0
-3 5 0
-5 3 0
1 -4 0
-3 1 0
2 0 c restrição adicionada
-1 0 c restrição adicionada
```

O resultado obtido pelo solver minisat é UNSAT, logo a afirmação é **correta**.

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

   No formato CNF, esta afirmação representa-se do seguinte modo:

   `¬p5`

   Para obter resposta a esta questão é necessário que: `Γ ⊨ ¬p5`, ou seja, que `F ⇒ ¬p5` seja válido.

```
F ⇒ ¬p5 é equivalente a ¬F V ¬p5 e para provar a sua validez é necessário provar
que ¬(¬F V ¬p5) é UNSAT.

Assim, queremos provar que F ∧ ¬(¬p5) é UNSAT, ou seja, que F ∧ p5 é UNSAT.
```

Para isso podemos recorrer ao SAT Solver minisat, escrevendo o seguinte ficheiro DIMACS CNF:

```
p cnf 5 8
-1 2 0
3 4 0
-2 -5 0
-3 5 0
-5 3 0
1 -4 0
-3 1 0
5 0 c restrição adicionada
```

O resultado obtido pelo solver minisat é UNSAT, logo a afirmação é **correta**.