

Professora Inês Dutra

Tópicos Avançados em Inteligência Artificial

Sexta Entrega: Representação de Conhecimento e Utilização
de Razão Utilizando *ProbLog*

Outubro de 2022

Trabalho realizado por:

Pedro Leite - 201906697

1. Exercício 1

```
1 0.4 :: heads.
2 0.3 :: col(1,red); 0.7 :: col(1,blue).
3 0.2 :: col(2,red); 0.3 :: col(2,green); 0.5 :: col(2,blue).
4 win :- heads, col(_,red).
5 win :- col(1,C), col(2,C).
```

a. Computar a probabilidade de ganhar o jogo

```
8 query(win).
```

Query ▼	Location	Probability
win	8:7	0.562

A *query* calcula a probabilidade do argumento que representa a vitória do jogo.

b. Computar a probabilidade de ter uma bola vermelha

```
11 red :- col(_,red).
12 query(red).
```

Query ▼	Location	Probability
red	12:7	0.44

Definimos uma variável que representa todas as bolas vermelhas. E calculamos com a *query*, a probabilidade dessa variável.

c. Computar a probabilidade de ganhar um jogo se a moeda for cara

```
16 evidence(heads).
17 query(win).
```

Query ▼	Location	Probability
win	17:7	0.79

O *evidence* define que a moeda vai ser caras. Tendo isto em conta, calculamos com a *query*, a probabilidade de ser vitória.

d. Computar a probabilidade de a primeira bola ser azul, sendo que ganhamos o jogo

```
20 evidence(win).
21 query(col(1,blue)).
```

Query ▼	Location	Probability
col(1,blue)	21:7	0.72241993

O *evidence* define que ganhamos. Tendo isto em conta, calculamos com a *query*, a probabilidade da primeira bola ser azul.

e. Computar a probabilidade de a primeira bola ser azul, sendo que perdemos o jogo

```
24 evidence(not win).
25 query(col(1,blue)).
```

Query ▼	Location	Probability
col(1,blue)	25:7	0.67123288

O *evidence* define que não ganhamos, através da negação *not*. Tendo isto em conta, calculamos com a *query*, a probabilidade da primeira bola ser azul.

f. Computar a probabilidade que qualquer bola é azul, sendo que ganhamos o jogo

```
28 evidence(win).
29 blue :- col(_,blue).
30 query(blue).
```

Query ▼	Location	Probability
blue	30:7	0.82918149

O *evidence* define que ganhamos. Tendo isto em conta, definimos uma variável que representa todas as bolas azuis e calculamos com a *query*, a probabilidade de qualquer bola ser azul.

2. Exercício 2

```
1 0.7 :: burglary.
2 0.2 :: earthquake.
3
4 0.9::alarm :- burglary, earthquake.
5 0.8::alarm :- burglary, \+earthquake.
6 0.1::alarm :- \+burglary, earthquake.
7 0.0::alarm :- \+burglary, \+earthquake.
```

Definimos a probabilidade de roubo como 0.7 e de terramoto como 0.2, de roubo e terramoto como 0.9, de apenas roubo como 0.8, de apenas terramoto como 0.1, de nenhum como 0. Utilizamos “\+” para definir que o roubo ou que o terramoto não acontece.

a. Qual é a probabilidade do alarme disparar?

```
10 query(alarm).
```

Query ▼	Location	Probability
alarm	10:7	0.58

A *query* calcula a probabilidade do alarme disparar.

b. Quando o alarme dispara, qual é a probabilidade que um roubo tenha ocorrido?

```
13 evidence(alarm).  
14 query(burglary).
```

Query ▼	Location	Probability
burglary	14:7	0.98965517

O *evidence* define que o alarme dispara. Tendo isto em conta, calculamos com a *query*, a probabilidade de um roubo ter ocorrido.

c. Quando o alarme dispara, qual é a probabilidade que um terremoto tenha ocorrido?

```
17 evidence(alarm).  
18 query(earthquake).
```

Query ▼	Location	Probability
earthquake	18:7	0.22758621

O *evidence* define que o alarme dispara. Tendo isto em conta, calculamos com a *query*, a probabilidade de um terremoto ter ocorrido.

3. Exercício 3

```
1 0.5::heads1.  
2 0.5::heads2.  
3  
4 two_heads :- heads1, heads2.  
5  
6 query(two_heads).
```

Query ▼	Location	Probability
two_heads	6:7	0.25

Inicialmente só tinha sido definida uma variável para a moeda, mas para este caso é necessário definir duas variáveis, uma para quando lançarmos a moeda pela primeira vez, e outra para quando lançarmos a moeda pela segunda vez. Utilizamos a *query* para calcular a probabilidade, do argumento que engloba estes dois lançamentos. Que vai ser 0.25, como pretendido.