

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

GABRIEL RIBEIRO BERNARDI - 11821BCC036

AARE - Avaliação Parcial da Aula 2

UBERLÂNDIA, MG

2020

Exercício 1

```
rico(a).  
rico(b).  
rico(c).  
rico(d).  
rico(e).  
  
famoso(k).  
famoso(l).  
famoso(m).  
famoso(n).  
famoso(o).  
  
eh_feliz(X) :- rico(X); famoso(X).
```

Esse exercício retorna as pessoas que são felizes, sejam elas ricas ou famosas.

Exercício 2

```
barb(a) .  
barb(b) .  
barb(c) .  
barb(d) .  
  
vai barbeiro(X) :- not(barb(X)) .
```

Nesse código, temos que quem vai ao barbeiro é quem não barbeia-se. Para que possamos fazer isso, usamos o predicado `not/1` em que fará a negação de `barb(X)`.

Exercício 3

[illegible]

```
H2 \= V1, H2 \= V2,  
V1 \= V2.
```

Para solucionar esse problema, temos que pensar em algo parecido com o exercício da cruzada. Verificamos as letras em que tem que ser comum para que a cruzadinha seja concluída.

Exercício 4

```
ladrao(xandao).  
  
gosta(lucas, arroz).  
gosta(gabriel, feijao).  
  
gosta(xandao, feijao).  
  
poderoubar(X, Y) :- ladrao(X), gosta(X, Z), gosta(Y, Z).
```

Nessa questão, temos que para que alguém (A1) possa roubar outro alguém (A2), é necessário que A1 seja ladrão e que ele goste da mesma coisa que A2 goste.

Exercício 5

```
influencer(penny).  
influencer(mary).  
influencer(john).  
  
likes(john,mary).  
likes(john,ringo).  
likes(mary,john).  
likes(john,beer).  
likes(john,wine).  
likes(mary,wine).  
likes(ringo,john).  
likes(ringo,beer).  
likes(george,mary).  
  
likes_(X, Y) :- not(influencer(X)), influencer(Z), influencer(W), Z \=  
W, likes(Z, Y), likes(W, Y), !.
```

Por fim, nessa questão temos que para satisfazer a função likes_, temos que ter alguém (A1) que goste de algo, desde que A1 não seja influencer e que existam pelo menos outras duas pessoas (A2) e (A3), sejam influencers e que gostem desta coisa.