



## **Lista 11: Linguagens de Programação: Programação Lógica**

### **Eng. de Computação – Belo Horizonte**

Prof. Andrei Rimsa Alvares

Aluno: Alexandre Roque.

### **Sumário**

Questão 1: .....	2
Questão 2: .....	2
Questão 3: .....	3
Questão 4: .....	4
Questão 5: .....	5

### Questão 1:

**Exercício 01)** Verifique se os seguintes termos podem ser unificados, se puderem, informe o resultado da unificação.

- a)  $p(X, 3)$  e  $q(2, 3)$
- b)  $p(X, 3)$  e  $p(2, 3)$
- c)  $p(x, 3)$  e  $p(2, 3)$
- d)  $p(X, 3)$  e  $p(Y, Y)$
- e)  $filho(carlos, maria)$  e  $filho(Y, Y)$
- f)  $[1, 2 \mid X]$  e  $[1, 2, 7, 3, 4]$
- g)  $[1, 2 \mid X]$  e  $[1, 2]$

1) a) Não é unificado (Nomes diferentes)

b)  $X = 2$

c) Não é unificado (x com letra minúscula)

d)  $X = Y, Y = 3$

e) Não é unificado ( $carlos \neq maria$ )

f)  $X = [7, 3, 4]$

g)  $X = []$

### Questão 2:

**Exercício 02)** Considere o seguinte programa em Prolog e uma consulta:

```
gosta(maria, peixe).
gosta(pedro, vinho).    ?- gosta(maria, X), gosta(pedro, X).
gosta(maria, vinho).    X = vinho .
gosta(pedro, maria).
```

Para essa consulta, mostre as árvores de execução do algoritmo de unificação quando:

- a) Um termo não pôde ser unificado, logo antes de iniciar o *backtracking*.
- b) A consulta foi unificada com sucesso e retornou a resposta.

2) a)

$gosta(maria, X)$

$gosta(pedro, X)$

$X = peixe$

$X = peixe$

$gosta(maria, peixe)$

false

b)

gosta(maria, X)

gosta(pedro, X)

X = vinho

X = vinho

gosta(maria, vinho)

gosta(pedro, vinho)

X = vinho

### Questão 3:

**Exercício 03)** Suponha que você conhece as velocidades médias de vários automóveis em uma determinada pista de corrida e a quantidade de tempo que eles permanecem na pista. Codifique esta informação nos fatos: velocidade e tempo. Em seguida crie uma cláusula chamada **distancia** para calcular a relação entre a velocidade e o tempo. Por exemplo:

```
? distancia(fusca, R).  
2005 .
```

velocidade(fusca, 115).

velocidade(corsa, 90).

velocidade(ranger, 120).

tempo(fusca, 45).

tempo(corsa, 60).

tempo(ranger, 35).

```
distancia(X, R) :- tempo(X, T),  
                   velocidade(X, S),  
                   R is S*T.
```

#### Questão 4:

**Exercício 04)** Escreva uma descrição Prolog de sua árvore genealógica (retrocedendo até seus avós e incluindo todos os descendentes) baseando somente nos fatos: **pai** e **mae**. Em seguida, escreva um conjunto de regras para as relações familiares: **avos**, **pais**, **irmaos**, **tios** e **primos**. Dica: use a consulta **findall(Object, Goal, List)** que produz uma lista **List** de todos os objetos **Object** que satisfazem a meta **Goal**. Por exemplo, para obter os nomes de todos os avós de Carlos.

```
?- avos(carlos, R).  
R = [joao, jose, maria, ana] .
```

```
pai(domingos, sonia).  
pai(domingos, sandro).  
pai(domingos, sergio).  
mae(maria, sonia).  
mae(maria, sandro).  
mae(maria, sergio).  
pai(ernandesAvo, ernandes).  
pai(ernandesAvo, reginaAmelia).  
pai(ernandesAvo, joseRoberto).  
pai(ernandesAvo, magali).  
pai(ernandesAvo, marcio).  
pai(ernandesAvo, zePequeno).  
pai(ernandesAvo, reginaCelia).  
pai(ernandesAvo, luiz).  
pai(ernandesAvo, auxiliadora).  
mae(cirene, ernandes).  
mae(cirene, reginaAmelia).  
mae(cirene, joseRoberto).  
mae(cirene, magali).  
mae(cirene, marcio).  
mae(cirene, zePequeno).  
mae(cirene, reginaCelia).  
mae(cirene, luiz).  
mae(cirene, auxiliadora).  
pai(ernandes, alexandre).  
pai(ernandes, ana).  
pai(ernandes, andrea).  
mae(sonia, alexandre).  
mae(sonia, ana).  
mae(sonia, andrea).  
familia(X, P) :- pai(X, P),  
                 mae(X, P).  
avo(X, P)      :- familia(R, P),  
                 familia(X, R).  
irmao(X, P)    :- pai(R, X),  
                 pai(R, P),
```

```

        mae(M, X),
        mae(M, P),
        X\=P.
tio(X, P) :-
    familia(F, P),
    irmao(F, X).

avos(X, R)      :- findall(Y, avo(Y, X), R).
pais(X, R)      :- findall(Y, familia(Y, X), R).
irmaos(X, R)    :- findall(Y, irmao(Y, X), R).
tios(X, R)      :- findall(Y, tio(Y, X), R).

```

## Questão 5:

**Exercício 05)** Considere o seguinte banco de dados com dados de clientes de uma determinada loja que vende smartphones.

Código	Nome	Idade	Profissão	Produto
01	Gabriel	21	Estudante	Iphone 4s por R\$1300,00, Galaxy S3 por R\$1400,00
02	Bruna	30	Médico	Iphone 5s por R\$2400,00, Capa Iphone 5s por R\$80,00
03	Rafaela	25	Arquiteto	Iphone 5s por R\$2500,00, Galaxy S4 por R\$2300,00
04	Victor	39	Advogado	BlackBerry Z10 por R\$1800,00
05	Beatriz	18	Estudante	Iphone 4s por R\$1300,00, Iphone 4 por R\$1000,00
06	João	28	Engenheiro	Nokia Lumia por R\$1700,00

- Para cada uma das linhas do banco de dados, crie fatos em Prolog.
- Defina cláusulas para consultar as seguintes informações:
  - listar\_clientes(L)**: obter a lista de todos os nomes dos clientes (L)
  - listar\_dados\_cliente(Codigo, Dados)**: obter todos os dados do cliente com exceção do código (Codigo) para um determinado cliente (Codigo)
  - listar\_smartphones(L)**: listar o nome (único) de todos os smartphones vendidos pela loja (L).
  - listar\_estudantes(L)**: listar o nome de todos os clientes da loja que sejam estudantes (L).
  - preco\_medio(X)**: obter o preço médio de todos os smartphones vendidos na loja (X).

a)

```

client(01, gabriel, 21, estudante).
client(02, bruna, 30, medico).
client(03, rafaela, 25, arquiteto).
client(04, victor, 39, advogado).
client(05, beatriz, 18, estudante).
client(06, joao, 28, engenheiro).
order(01, iphone4s, 1300).
order(01, galaxys3, 1400).
product(iphone4s).
order(02, iphone5s, 2400).
product(galaxys3).
order(02, ciphone5s, 80).
product(iphone5s).

```

```
order(03, iphone5s, 2500).  
product(ciphone5s).  
order(03, galaxys4, 2300).  
product(galaxys4).  
order(04, blackberryz10, 1800).  
product(blackberryz10).  
order(05, iphone4s, 1300).  
product(iphone4).  
order(05, iphone4, 1000).  
product(nokialumia).  
order(06, nokialumia, 1700).
```

b)

i. listar\_clientes(L) :- findall(N, client(\_, N, \_, \_), L).

ii. listar\_dados\_cliente(Codigo, Dados) :- findall([N, I, P], client(Codigo, N, I, P), Dados).

iii. listar\_smartphones(L) :- findall(Nome, product(Nome), L).

iv. listar\_estudantes(L) :- findall(Nome, cliente(\_, Nome, \_, estudante), L).