

Fundamentos

Marco A L Barbosa

malbarbo.pro.br

Os exercícios sem referências estão licenciados com uma Licença Creative Commons - Atribuição-CompartilhaIgual 4.0 Internacional.



<https://github.com/malbarbo/na-proglog>

1. [lpn 2.1] Quais dos seguintes pares de termos unificam? Quando for o caso, dê o valor instanciado para cada variável que levou a unificação.

- a. `bread = bread`
- b. `'Bread' = bread`
- c. `'bread' = bread`
- d. `Bread = bread`
- e. `bread = sausage`
- f. `food(bread) = bread`
- g. `food(bread) = X`
- h. `food(X) = food(bread)`
- i. `food(bread, X) = food(Y, sausage)`
- j. `food(bread, X, beer) = food(Y, sausage, X)`
- k. `food(bread, X, beer) = food(Y, kahuna_burger)`
- l. `food(X) = X`
- m. `meal(food(bread), drink(beer)) = meal(X, Y)`
- n. `meal(food(bread), X) = meal(X, drink(beer))`

2. [lpn 1.3] Quantos fatos, regras, cláusulas e predicados existem na seguinte base de conhecimento? Qual é a cabeça de cada regra e quais são as metas que elas contêm?

```
mulher(vincent).
mulher(mia).
homem(jules).
pessoa(X) :- homem(X); mulher(X).
ama(X, Y) :- pai(X, Y).
pai(Y, Z) :- homem(Y), filho(Z, Y).
pai(Y, Z) :- homem(Y), filha(Z, Y).
```

3. Dado a seguinte base de conhecimento

```
pai(adao, abel).
pai(adao, caim).
pai(adao, sete).
pai(caim, enoque).
pai(enoque, irad).
pai(irad, meujael).
pai(meujael, metusael).
pai(metusael, lameque).
pai(lameque, jabal).
pai(lameque, jubal).
pai(lameque, tubalcaim).
pai(lameque, naama).
```

```
mae(eva, abel).
mae(eva, caim).
```

```
mae(eva, sete).
mae(ada, jabal).
mae(ada, jubal).
mae(zila, tubalcaim).
mae(zila, naama).
```

```
homem(sete).
homem(caim).
homem(jabal).
homem(jubal).
homem(tubalcaim).
```

```
mulher(naama).
```

defina (e teste) as seguintes regras em Prolog:

- a. X é homem se X é pai.
- b. X é mulher se X é mãe.
- c. X e Y são irmãos se X e Y têm o mesmo pai ou mesma mãe.
- d. X e Y são casados se eles têm um filho.
- e. avo(X, Y), X é avó de Y.
- f. irma(X, Y), X é irmã Y.
- g. irmao(X, Y), X é irmão Y.
- h. e_pai(X), X é pai.
- i. e_mae(X), X é mãe.
- j. e_filho(X), X é filho.
- k. ancestral(X, Y), X é ancestral de Y.

4. Projete um predicado para encontrar o valor de cada letra no seguinte enigma

```
SEND +
MORE =
MONEY
```

de maneira que a adição fique correta. O valor de cada letra deve ser um dígito e todas as letras devem ter valores distintos. A letra M não poder ser 0. Note que a posição que a letra aparece muda o valor que ela representa no número. Por exemplo, se $E = 3$, no “número” SEND a letra E corresponde a 300 enquanto que no “número” MORE ela corresponde a 3 e em MONEY 30. Use restrições sobre inteiros!

5. (Desafio) [pp99 2.01] Projete um predicado `primo(+X)` que é verdadeiro se X é um número primo. Exemplo

```
?- primo(7).
true.
```

```
?- primo(4).
false.
```

6. (Desafio) Projete um predicado `perfeito(+X)` que é verdadeiro se X é um número perfeito. Um número é perfeito se a soma dos seu divisores próprios é igual a ela. Por exemplo, o número 6 é perfeito, pois $6 = 1 + 2 + 3$. O número 28 também é perfeito, pois $28 = 1 + 2 + 4 + 7 + 14$.

```
?- perfeito(6).
true.
```

```
?- perfeito(9).
false.
```

Referências

- [pp99]. 99 problemas para resolver em (Prolog)
- [lpn]. Lear Prolog Now
- [pip]. Programming in Prolog.