

Inteligência Artificial
Lista de Exercícios
Assunto: Predicado de Controle

1ª. O programa a seguir associa a cada pessoa seu esporte preferido.

joga(ana,volei).

joga(bia,tenis).

joga(ivo,basquete).

joga(eva,volei).

joga(leo,tenis).

Suponha que desejamos consultar esse programa para encontrar um parceiro P para jogar com Leo. Então, podemos realizar essa consulta de duas formas:

a) ?- jogar(P,X), jogar(leo,X), P\=leo.

b) ?- jogar(leo,X), jogar(P,X), P\=leo.

Desenhe as árvores de busca construídas pelo sistema ao responder cada uma dessas consultas. Qual consulta é mais eficiente, por quê?

2ª. O predicado num classifica números em três categorias: positivos, nulo e negativos. Esse predicado, da maneira como está definido, realiza retrocesso desnecessário. Explique por que isso acontece e, em seguida, utilize cortes para eliminar esse retrocesso.

num(N,positivo) :- N>0.

num(0,nulo).

num(N,negativo) :- N<0.

3ª. Suponha que o predicado fail não existisse em Prolog. Qual das duas definições a seguir poderia ser corretamente usada para causar falhas?

a) falha :- (1=1).

b) falha :- (1=2).

Considere o programa a seguir:

animal(cão).

animal(canário).

animal(cobra).

animal(morcego).

animal(gaivota).

voa(canário).

voa(morcego).

voa(gaivota).

dif(X,X) :- !, fail.

dif(_,_).

pássaro(X) :- animal(X), voa(X), dif(X,morcego).

Desenhe a árvore de busca necessária para responder a consulta

?- pássaro(X).

Em seguida, execute o programa para ver se as respostas do sistema correspondem àquelas que você encontrou.

4ª. O que está errado no programa abaixo? Rode-o com trace, para ? max(3,4,M) e ?- max(4,3,M).

```
max(X,Y,M):-!, X>Y, M=X.  
max(X,Y,M):-!, X<=Y, M=Y.
```

5ª. Seja o seguinte programa Prolog:

```
p(1).  
p(2) :- !.  
p(3).
```

Escreva todas as respostas do sistema Prolog para as seguintes consultas:

- (a) ?-p(X).
- (b) ?-p(X), p(Y).
- (c) ?-p(X), !, p(Y).

6ª. seguinte relação classifica números em três classes: positivo, nulo ou negativo:

```
classe(N, positivo) :- N > 0.  
classe(0, nulo).  
classe(N, negativo) :- N < 0.
```

Defina este procedimento de uma forma mais eficiente usando cuts.