Lista de Exercícios

Lógica em Programação

1) Uma pilha de blocos pode ser descrita por um conjunto de fatos do tipo *sobre(Bloco1, Bloco2)*, que são verdadeiros se *Bloco1* está sobre o *Bloco2*. Defina um predicado *acima(Bloco1, Bloco2)* que é verdadeiro se *Bloco1* está acima do *Bloco2* na pilha. (Dica: *acima* é o fecho transitivo de *sobre*.)

2) Escreva um programa Prolog para percorrer uma árvore em forma pós-fixa. (Para percorrer uma árvore em forma pós-fixa, primeiro se visita a sub-árvore da esquerda, depois a sub-árvore da direita, e por último a raiz.)

3) Analise a execução do seguinte programa:

```
reverse(Xs,Ys) :- reverse(Xs,[],Ys).

reverse([X|Xs],Revs,Ys) :- reverse(Xs,[X|Revs],Ys).
reverse([],Ys,Ys).
```

Esta é uma variação do problema *reverse*. A técnica utilizada neste programa é chamada *bottom-up*.

Utilize a consulta ?- reverse([a,b,c],RevList).

4) Escreva um programa Prolog para encontrar o minimo de uma lista de inteiros.

5) Analise a diferença entre os programas 1 e 2 abaixo: (Select(X,HasX,OneLessXs): A lista OneLessXs é o resultado da remoção de uma ocorrência de X na lista HasX.)

```
1) select(X,[X|Xs],Xs).
   select(X,[Y|Ys],[Y|Zs]) :- select(X,Ys,Zs).

2) select(X,[X|Xs],Xs) :- !.
   select(X,[Y|Ys],[Y|Zs]) :- select(X,Ys,Zs).
```

6) Dois jogadores por sua vez dizem um número entre 1 e 3 inclusive. A soma dos números é guardada, e o jogador que conseguir somar 20 primeiro ganha o jogo. Escreva um programa Prolog para jogar e ganhar, usando funções "memo". (Obs: funções "memo" são predicados Prolog adicionados ao banco de dados Prolog através de *assert*. Lembram-se de Ackermann?)

7) Escreva um programa que resolva o "problema do casamento estável" (Sedgewick, 1983), expresso da seguinte forma:

Assuma que N homens e N mulheres querem se casar entre eles. Cada homem tem uma lista de todas as mulheres na sua ordem de preferência, e cada mulher tem uma lista de todos os homens na sua ordem de preferência. O problema é encontrar um conjunto de casamentos estáveis.

Um conjunto de casamentos é *instável* se duas pessoas que não são casadas têm preferências cruzadas. Por exemplo, suponha que existem dois homens, A e B, e duas mulheres, X e Y, tais que A prefere X a Y, B prefere Y a X, X prefere A a B, e Y prefere B a A. O par de casamentos A-Y e B-X é instável, porque A prefere X a Y, enquanto X prefere A a B.

Seu programa deve ter como entrada listas de preferências, e como saída um conjunto de casamentos estáveis, isto é, um conjunto de casamentos que não sejam instáveis. Existe um teorema em teoria dos grafos que diz que é sempre possível encontrar uma solução estável. Teste seu programa com as seguintes listas de homens e mulheres com suas respectivas preferências:

```
avraham: chana tamar zvia ruth sarah
binyamin: zvia chana ruth sarah tamar
chaim: chana ruth tamar sarah zvia
david: zvia ruth chana sarah tamar
elazar: tamar ruth chana zvia sarah

zvia: elazar avraham david binyamin chaim
chana: david elazar binyamin avraham chaim
ruth: avraham david binyamin chaim elazar
sarah: chaim binyamin david avraham elazar
tamar: david binyamin chaim elazar avraham
```

8) Escreva um programa para resolver o seguinte problema lógico: existem 5 casas, cada uma com uma cor diferente e cada uma habitada por uma pessoa de nacionalidade diferente, com um animal de estimação, um tipo de bebida preferida e uma marca de cigarros preferida.

1. O inglês mora na casa vermelha.
2. O espanhol tem um cachorro.
3. Na casa verde se bebe café.
4. O ucraniano gosta de chá.
5. A casa verde é vizinha da direita da casa de cor marfim.
6. O fumador de Marlboro cria lesmas.
7. Se fuma Continental na casa amarela.
8. Na casa do meio se bebe leite.
9. O norueguês mora na primeira casa à esquerda.
10. A pessoa que fuma Free mora na casa ao lado da pessoa que tem uma raposa.
11. Na casa ao lado da casa onde há um cavalo, se fuma Continental.
12. A pessoa que gosta de fumar Hollywood também gosta de suco de laranja.
13. O japonês fuma Minister.
14. O norueguês mora na casa azul.

Quem é o dono da zebra?

9) Exercício (iii) da página 235, The Art of Prolog, Ehud Shapiro.

10) Escreva um programa para traduzir uma sentença escrita em DCG (Definite Clause Grammar) para uma cláusula Prolog.

11) Escreva um meta-interpretador que conte o número de chamadas a procedimentos feitos em um programa Prolog.