Lógica em Programação

Lista de Exercícios

1) Faça uma **breve** comparação entre linguagens declarativas (p.e., Prolog) e linguagens imperativas (p.e., C), destacando os pontos mais importantes.

2) Escreva de forma concisa o algoritmo de execução de Prolog.

3) Considere o seguinte programa:

```
top(X,Y) :- p(X,Y).
top(X,X) :- s(X).
p(X,Y) :- q(X), r(Y).
p(X,Y) :- s(X), r(Y).
```

```
q(a)
q(b).
r(c).
r(d).
s(e).
```

Faça um esboço da árvore de execução para a consulta ?- top(X,Y) com as respectivas soluções para cada ramo da árvore.

4) Transforme as seguintes cláusulas DCG para as suas equivalentes em Prolog:

```
cl --> g1, g2, g3.
termo --> [maria].
```

5) Considere o seguinte programa:

```
p(Y) :- q(X,Y), r(Y).
p(X) :- q(X,X).
q(a,a).
q(a,b).
r(b).
```

Introduza o operador de corte (cut, !) em lugares diferentes dete programa e diga quantas e quais respostas você obteria para a consulta `p(Z)`.

6) Qual a função do programa abaixo?

```
c(_, [], 0).
c(X, [X|L], N):-
```

```
        c(X, L, R), !,
        N is R+1.
c(X, [_|L], N):-
        c(X, L, N).
```

7) Escreva um programa em Prolog para fazer multiplicação de matrizes. Obviamente, o programa falha se o número de colunas da primeira matriz não for igual ao número de linhas da segunda matriz.

8) Escreva um programa em Prolog que implemente o algoritmo de unificação.

9) Por causa de um 'bug' no compilador que gera código de Prolog para WAM, o código gerado para o programa abaixo está incompleto e com erros.

a) Descubra quais são as instruções que estão faltando e os erros.

b) O que este programa faz?

c) Crie uma instrução ou um conjunto de instruções que reduza o número total de instruções necessárias para selecionar uma cláusula alternativa para execução. Modifique o código WAM gerado para incluir a(s) instrução(ões) criada(s) (dica: instruções de indexação).

Programa fonte Prolog:

```
pred(X,[X]).
pred(X,[_|Y]) :- pred(X,Y).
```

Código WAM gerado:

```
pred_2:         try_me_else pred_2_2
pred_2_1:       get_variable X1,A1
                proceed
pred_2_2:       get_variable X1,A1
                put_value X1,A1
                put_value X2,A2
                execute pred_2
```

10) O programa abaixo foi escrito em Parlog. Quais são as diferenças básicas entre a execução deste programa em um sistema Parlog e em um sistema Prolog sequencial? Comente sobre diferenças sintáticas e semânticas.

```
mode sort(?,^), qsort(?,^,?), partition(?,?,^,^).
```

```
sort(List,Sorted) <- qsort(List,Sorted,[]).

qsort([X|Y],Sorted1,Sorted2) <-
     partition(Y,X,L1,L2),
     qsort(L1,Sorted1,[X|Sorted]),
     qsort(L2,Sorted,Sorted2).
qsort([],S,S).

partition([H|T],X,[X|L1],L2) <- X < H :
     partition(T,X,L1,L2).
partition([H|T],X,L1,[X|L2]) <- H =< X :
     partition(T,X,L1,L2).
```

11) Dado o programa Prolog abaixo e considerando-se que **portabilidade** é mais importante que **desempenho/aceleração**, como este programa poderia ser executado em paralelo? Qual seria o tipo de arquitetura mais conveniente para a execução paralela deste programa (comente sobre localidade, escalabilidade, topologia, aceleração e quantidade de paralelismo)?

```
paper(P,D,I) :- date(P,D), author(P,A), loc(A,I,D).
paper(P,D,I) :- tr(P,I), date(P,D).
paper(xform,1978,uci).

author(fp,backus).
author(df,arvind).
author(eft,kling).
author(pro,pereira).
author(sem,vanemdem).
author(db,warren).
author(sasl,turner).
author(xform,standish).

date(fp,1978).
date(df,1978).
date(eft,1978).
date(pro,1978).
date(sem,1976).
date(db,1981).
date(sasl,1979).

title(db,efficient_processing_of_interactive_etc).
title(df,an_synchronous_programming_language_etc).
title(eft,value_conflicts_and_social_choice_etc).
title(fp,can_programming_be_liberated_etc).
title(pro,dec_10_prolog_user_manual).
title(sasl,a_new_implementation_technique_etc).
```

```
title(sem,the_semantics_of_predicate_logic_etc).
title(xform,irvine_program_transformation_catalog).

loc(arvind,mit,1980).
loc(backus,ibm,1980).
loc(kling,uci,1978).
loc(pereira,lisbon,1978).
loc(vanemdem,waterloo,1980).
loc(turner,kent,1981).
loc(warren,edinburgh,1977).
loc(warren,sri,1982).

journal(fp,cacm).
journal(sasl,spe).
journal(kling,cacm).
journal(sm,jacm).

tr(db,edinburgh).
tr(df,uci).
```

12) Escreva um programa em Prolog que retorne os 'n' primeiros termos de uma série de Fibonacci e sua soma, i.e., `?- fib(3,L,S)` deveria retornar `L = [0,1,1]` e `S = 2`.

13) Quais são as características básicas que diferenciam Prolog de uma linguagem que implementa programação lógica com restrições? Exiba um programa Prolog que não falha ou não dá mensagens de erro quando executado em um sistema que implementa programação lógica com restrições.