

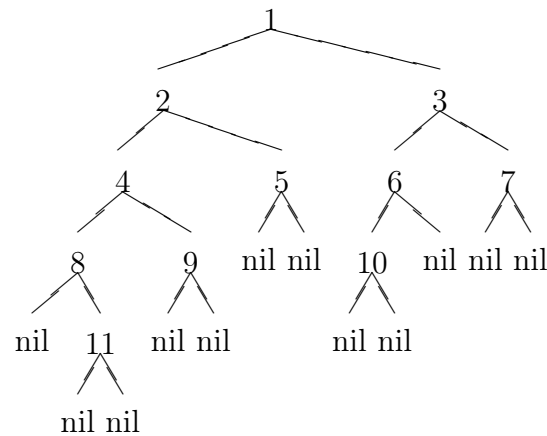
Departamento de Ciência da Computação - ICEX - UFMG
 Disciplina: Inteligência Artificial
 Professor: José de Siqueira
 2º Semestre de 2011

1ª Prova

Data de entrega: 12/9/2011

Valor: 18

1- (6 pts.) Dada a árvore abaixo,



há 3 maneiras de percorrê-la:

- Percurso simétrico: 8, 11, 4, 9, 2, 5, 1, 10, 6, 3, 7.
- Percurso préfixado: 1, 2, 4, 8, 11, 9, 5, 3, 6, 10, 7.
- Percurso pósfixado: 11, 8, 9, 4, 5, 2, 10, 6, 7, 3, 1.

a) Faça um programa Prolog que percorra a árvore, imprimindo na tela a sequência dos nós para cada um dos três tipos de percurso, dada a seguinte representação para a árvore acima:

```
[1, [2, [4, [8, [], 11], 9], 5], 3, [6, [10], 7]]
```

b) Proponha uma representação para a árvore de modo que os percursos sejam feitos como no seguinte programa Prolog:

```

simetrico([]). % Arvore vazia
simetrico([Folha]):- % Folha
    !,
    visita(Folha).
simetrico([Raiz,SubarvEsq,SubarvDir]):- % Arvores completas
    simetrico(SubarvEsq),
    visita(Raiz),
    simetrico(SubarvDir).
  
```

- c) Proponha uma representação para árvores que seja totalmente homogênea, de forma a eliminar a 2ª cláusula dos programas de percurso de árvore (veja o programa de percurso simétrico do item acima). Faça programas para os 3 percursos para a representação proposta.
- d) A que conclusão você chega com relação ao dualismo representação/ algoritmo?

2- (5 pts.)

- a) Proponha uma representação para os nós de uma árvore n-ária que contenha um campo com inteiros positivos.
- b) Faça um programa Prolog que utilize a representação proposta e retorne um caminho entre a raiz e uma folha da árvore, de forma não determinística.
- c) Faça um programa Prolog que retorne o caminho com maior soma dos nós dentre **todos** os caminhos existentes da raiz a **todas** as folhas da árvore. Dica: utilize o predicado `findall` para encontrar todas as soluções.

3- (1 pt.) Escreva um programa Prolog, o mais otimizado possível, que faça a ordenação em ordem crescente de uma lista de inteiros, por inserção de um elemento qualquer da lista na posição apropriada de uma lista ordenada.

4- (4 pts.) Escreva programas Prolog para os seguintes problemas:

- a) Divida uma lista em duas partes. O tamanho da primeira parte da lista é dada. Não use predicados embutidos!

Exemplo:

```
?- divide([a,b,c,d,e,f,g,h,i,k],3,L1,L2).
L1 = [a,b,c]
L2 = [d,e,f,g,h,i,k]
```

- b) Elimine cópias consecutivas de elementos de uma lista. Se uma lista contém elementos repetidos eles devem ser substituídos por uma cópia única do elemento. A ordem dos elementos não deve ser mudada. Exemplo:

```
?- comprime([a,a,a,a,b,c,c,a,a,d,e,e,e,e],X).
X = [a,b,c,a,d,e]
```

- c) Melhore o seguinte programa, para que fique mais eficiente, explicando porque e como sua solução o melhora:

```
processador_de_listas([],R):-e(R).
processador_de_listas([H|T],R):-
    processador_de_listas(T,TR),
    f(H,TR,R).
```

d) Se os predicados $e/1$ e $f/3$ do programa acima são definidos como abaixo:

```
e(0).  
f(A,B,C):-C is A+B.
```

o programa `processador_de_listas/2` resultante irá somar os elementos de uma lista de inteiros.

Já a definição abaixo:

```
e(0/0).  
f(X,A/B,A1/B1):-A1 is A+X, B1 is B+1.
```

pode ser usada para computar a média dos valores da lista de inteiros, para o mesmo programa `processador_de_listas/2` acima. Explique como e porquê.

5- (2 pts.) Considere o programa abaixo:

```
max([X],X).  
max([X|L],X):-max(L,V),X>=V,!.  
max([X|L],V):-max(L,V).
```

- O que faz o programa dado? Para quais modos de uso?
- Critique o programa acima, apontando suas deficiências.
- Proponha uma versão melhorada para o mesmo programa, introduzindo pelo menos duas melhorias, explicando-as.