Departamento de Ciência de Computadores - FCUP

Sistemas Inteligentes

Exame época normal 2013/14 (duração: 1 hora) **PARTE 2**

Nome: _____ Data: 20/06/2014

1. Qual é o resultado da aplicação das substituições abaixo na representação Prolog:

```
yes(F,L) :- append(F,c(L,nil),c(l,c(i,c(s,c(t,nil))))

% Substituições: {F/c(l,X1), Y1/c(L,nil), A1/l, Z1/c(i,c(s,c(t,nil)))}
```

   (a) `yes(c(l,X1),L) :- append(F,Y1,c(l,c(i,c(s,c(t,nil))))`
   (b) `yes(c(l,X1),L) :- append(c(l,X1),Y1,c(l,c(i,c(s,c(t,nil))))`
   (c) `yes(c(l,X1),A1) :- append(c(l,X1),Y1,c(l,c(i,c(s,c(t,nil))))`
   (d) `yes(c(A1,X1),A1) :- append(c(l,X1),Y1,c(l,c(i,c(s,c(t,nil))))`
   (e) `yes(c(A1,X1),L) :- append(c(l,X1),Y1,c(l,c(i,c(s,c(t,nil))))`
   (f) `yes(c(A1,X1),L) :- append(c(A1,X1),Y1,c(A1,Z1))`
   (g) Não é possível aplicar nenhuma substituição.

2. Qual seria a melhor conversão para Prolog da representação em DCG `s(X) --> t(X), u(X).` :

   (a) `s(X,Y,Z) :- t(X,Y,Z), u(X,Y,Z).`
   (b) `s(X,Y,Z) :- t(X,Y1,Y2), u(X,Y2,Z).`
   (c) `s(X,Y,Z) :- t(Y1,Y2,X), u(Y2,Z,X).`
   (d) `s(X,Y,Z) :- t(X,Y,Y1), u(X,Y1,Z).`
   (e) `s(X,Y,Z) :- t(Y,Y1,X), u(Y1,Z,X).`
   Nenhuma das respostas anteriores está correta.

3. Dada a expressão de ganho de informação do atributo A mostrada em 1, indique qual das tabelas melhor se adequa a este cálculo.

$$\frac{1}{7}I(1,0,0) + \frac{3}{7}I(\frac{2}{3},\frac{1}{3},0) + \frac{3}{7}I(0,\frac{1}{3},\frac{2}{3}) \quad (1)$$

(a)

A	Classe
v1	c1
v3	c3
v2	c1
v2	c2
v3	c2
v2	c1
v3	c3

(b)

A	Classe
v1	c1
v3	c1
v1	c1
v3	c2
v1	c2
v2	c1
v3	c2

(c)

A	Classe
v1	c1
v3	c2
v2	c1
v2	c2
v3	c1
v1	c2
v3	c2

4. No mundo dos blocos, dado o estado inicial:

```
%
%
%    +-+   +-+  +-+
%    |c|   |b|  |a|
%    +-+   +-+  +-+
%    |e|   |d|  |f|
%----+-+---+-+--+-+---------------- floor
```

   e o estado final:

```
%
%    +-+    +-+
%    |e|    |b|
%    +-+    +-+
%    |a|    |c|
%----+-+----+-+--------------------- floor
```

que tipo de ações poderia utilizar para construir um plano que leve um agente a chegar do estado inicial ao final:

(a) `goto(x,y)`, onde x é o lugar inicial e y é o lugar de destino. Com pré-condição: canGo(x,y) and at(x) e pós-condição: at(y) and notAt(x)

(b) `pickup(u,x)`, onde u é um bloco localizado em x. Com pré-condição: clean(u) and at(u,x) e pós-condição: at(u,OtherPlace)

(c) `putdown(u,x)`, onde u é um bloco depositado no local x. Com pré-condição: holding(u) and at(x) e pós-condição: notHolding(u) and at(x)

(d) `move(b,x,y)`, mover bloco b do local x para o local y. Com pré-condição: at(b,x) and clean(b) and clean(y) e pós-condição: on(y,b) and clean(x)

(e) Nenhum dos tipos anteriores me faz gerar um plano adequado para este problema.

5. Em um plano de ordem parcial:

(a) Todas a variáveis devem estar instanciadas.

(b) Todas as ações estão linearizadas.

(c) Algumas acões não estão linearizadas.

(d) Algumas variáveis estão instanciadas.

(e) Nada do que foi dito acima é verdadeiro.

6. Qual é a função de um algoritmo de indução de árvores de decisão?

(a) Gerar a árvore mais compacta possível.

(b) Gerar um padrão que descreve as observações.

(c) Gerar a árvore maior possível.

(d) Encontrar relações entre os atributos de uma tabela.

(e) Nenhuma das respostas anteriores descreve a função de um algoritmo de indução de árvores.

7. (a) Gerar a árvore mais compacta possível.

(b) Gerar um padrão que descreve as observações.

(c) Gerar a árvore maior possível.

(d) Encontrar relações entre os atributos de uma tabela.

(e) Nenhuma das respostas anteriores descreve a função de um algoritmo de indução de árvores.

8. O seguinte código Prolog implementa uma busca em profundidade, porém não está completamente correto. Por que? Reescreva este código de forma que se torne correto.

```
solve(N,[N]) :- goal(N).
solve(N,[N|Sol1]) :-
     s(N,N1),
     solve(N1,Sol1).
```