Ciência de Computadores

Sistemas Inteligentes

Segundo teste (duração: 2 horas)

Data: 12 de Junho de 2014

1) Qual seria a melhor representação em Prolog para a função que calcula o fatorial de um número inteiro?

(a)

```
fat(0,1).
fat(N,N*fat(N-1)).
```

(b)

```
fat(0,1).
fat(N,N*fat(N1)) :- N1 is N - 1.
```

(c)

```
fat(0,1).
fat(N,N*N1) :- N1 is fat(N-1).
```

(d)

```
fat(0,1).
fat(N,N1) :- N1 is N*fat(N-1).
```

(e)

```
fat(0,1).
fat(N,N1) :- fat(N-1,N2), N1 is N*N2.
```

(f)

```
fat(0,1).
fat(N,N1) :- N2 is N - 1, fat(N2,N3), N1 is N*N3.
```

2) O resultado da unificação entre os termos `menos(X,mais(Y,Z))` e `menos(3,menos(4,T))`, em Prolog, onde os nomes de variáveis são iniciados por letra maiúscula e as constantes por letra minúscula, é:

(a) `X=3, Y=4, Z=T`

(b) `X=3, Y=menos(4,T), Z=T`

(c) `X=3, Y=4, Z=menos(4,T)`

(d) `X=3, mais(Y,Z)=menos(4,T)`

(e) estes dois termos não são unificáveis.

---

3) Qual é a melhor interpretação para o programa abaixo:

```
p(X,Y) :- q(X,Y).

p(X,Y) :- p(X,Z), q(Z,Y).
```

(a) p é um predicado que implementa o conceito de progenitor.

(b) p é um predicado que implementa o conceito de antepassado.

(c) p é um predicado que implementa o conceito de avô.

(d) nenhuma das alternativas anteriores corresponde à interpretação de p.

---

4) Dada a expressão 1, calculada pelo algoritmo de indução de árvore de decisão, para um atributo numa tabela de observações, qual seria a representação mais apropriada para os valores deste atributo e da variável de interesse (classe)?

$$\frac{3}{6}I(\frac{2}{3},\frac{1}{3}) + \frac{1}{6}I(1,0) + \frac{2}{6}I(\frac{1}{2},\frac{1}{2}) \tag{1}$$

(a)

A	Classe
a1	c1
a2	c1
a3	c2
a1	c1
a2	c1
a3	c2

(b)

A	Classe
a1	c1
a1	c1
a1	c2
a2	c1
a3	c1
a3	c2

(c)

A	Classe
a1	c1
a1	c2
a2	c1
a2	c2
a3	c1
a3	c2

5) Num algoritmo para geração de planos que contém as ações abaixo, quais são os passos principais iniciais?

Start, com POSCOND: $on(A, B) \wedge on(B, C) \wedge clear(A)$

Finish, com PRECOND: $on(B, C) \wedge on(A, Table) \wedge clear(A) \wedge clear(B)$

(a) Selecionar cada uma das pós-condições do estado inicial (Start) e tentar traçar o plano utilizando ações do próprio plano ou do conjunto de operadores.

(b) Selecionar cada uma das pré-condições do estado final (Finish) e tentar traçar o plano utilizando ações do próprio plano ou do conjunto de operadores.

(c) Selecionar alternadamente uma pré-condição do estado final (Finish) e uma pós-condição (Start) até traçar um plano completo.

(d) Nenhuma das alternativas anteriores.

---

6) Sobre os comboios de Michalski, podemos afirmar que para encontrar um bom padrão de previsão para novos comboios:

(a) devemos representar cada comboio utilizando uma única linha de uma tabela.

(b) devemos representar cada comboio, de forma que cada grupo de linhas corresponda às características de um mesmo comboio.

(c) devemos dar preferência a uma representação relacional.

(d) Nenhuma das alternativas anteriores permite encontrar um bom padrão de previsão para os comboios.

---

7) Na implementação do agendamento de vôos com a sintaxe:

```
timetable(edinburgh,london,
[ 9:40/10:50/ba4733/alldays,
13:40/14:50/ba4773/alldays,
19:40/20:50/ba4833/[mo,tu,we,th,fr,su]]).
```

qual é a melhor definição de operadores que permite a correta unificação do termo 9:40/10:50/ba4733 com o termo X/Y/Z?

(a) `:- op(500,xfy,/). :- op(400,xfy,:).`

(b) `:- op(400,xfy,/). :- op(500,xfy,:).`

(c) `:- op(500,xfx,/). :- op(400,xfy,:).`

(d) Nenhuma das formas acima permite a unificação correta.

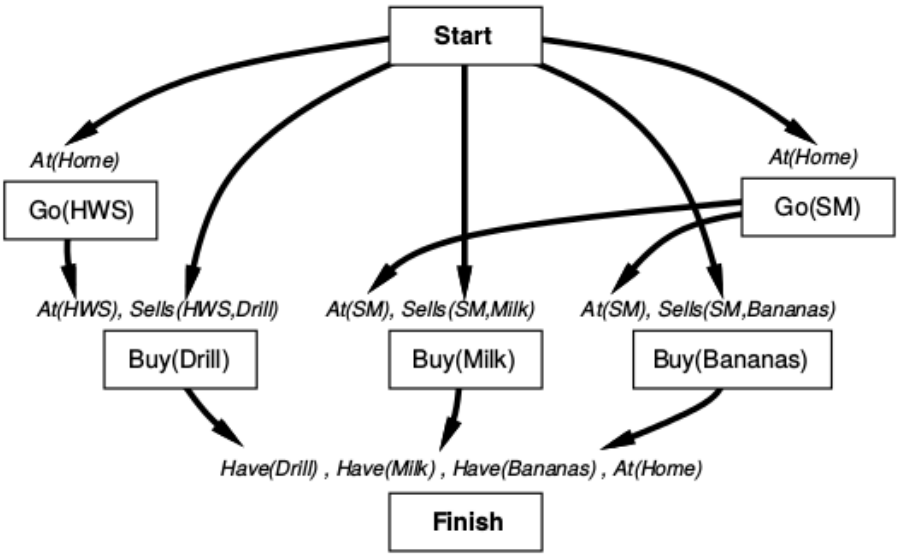

Figura 1: Figura para a questão 8

---

8) No plano de ordem parcial mostrado na Figura 1, retirada do livro, é necessário reordenar as duas ações `Go`. Por que esta reordenação é necessária?

(a) Para linearizar o plano e torná-lo executável.

(b) Para remover uma ameaça.

(c) Para simplificar o plano.

(d) Para tornar o plano completo.

(e) Para tornar o plano totalmente instanciado.

9) Dados os códigos Prolog abaixo, qual deles implementa corretamente a transformação de estados do problema da torre de Hanoi?

(a)

```
move(S1,[Origem,Destino|NewS]) :-
   delete(Origem,S1,NewS1),
   delete(Destino,NewS1,NewS).
```

(b)

```
move(S1,[Origem,[T|Destino]|NewS]) :-
   delete([T|Origem],S1,NewS1),
   delete(Destino,NewS1,NewS).
```

(c)

```
move(S1,[Destino,[T|Origem]|NewS]) :-
   delete([T|Origem],S1,NewS1),
   delete(Destino,NewS1,NewS).
```

(d)

```
move(S1,[[T1|Destino],[T2|Origem]|NewS]) :-
   delete([T1|Origem],S1,NewS1),
   delete([T2|Destino],NewS1,NewS).
```

(e)

```
move(S1,[[T|Destino],Origem|NewS]) :-
   delete(Origem,S1,NewS1),
   delete([T|Destino],NewS1,NewS).
```

(f) nenhuma das alternativas anteriores implementa corretamente esta transformação.