Paradigmas da Programação I
LECom (2º ano)

Exame de 1ª Época – 1ª Chamada

Data: 10 de Janeiro de 2005
Hora: 09:30

Dispõe de 2:30 horas para realizar este exame
Leia as questões com toda a atenção e responda com calma e clareza em folha convencional

Questão 1: Representação de Conhecimento

Recolheu-se informação sobre o sucesso escolar (a nível do ensino superior) e a idade e estado profissional dos alunos. Analisados os dados, apuraram-se as seguintes evidências:

- dos estudantes trabalhadores, que se matriculam até aos 30 anos, 40% chegam ao fim do curso.
- dos estudantes trabalhadores, que se matriculam com mais de 30 anos, só 10% chegam ao fim do curso.
- dos estudantes que se matriculam até aos 18 anos, 95% concluem o curso em 5 anos.
- dos estudantes que se matriculam até aos 18 anos, 5% não concluem o curso.
- dos estudantes retardatários, que se matriculam depois dos 50 anos, 75% não concluem o curso.
- dos estudantes retardatários, que se matriculam depois dos 50 anos, 75% dos que concluem o curso, já tinham alguma formação superior.
- dos estudantes que estão matriculados mais de 9 anos, 100% não chegam ao fim do curso.

Pretende-se que:

a) escreva um conjunto de cláusulas de Horn, em Prolog, que descreva o conhecimento extraído do estudo feito e acima referido.

b) mostre como poderia interrogar a sua BC (base de conhecimento), para saber qual a percentagem de alunos retardatários (mais de 50 anos) que concluem o curso.

c) mostre como poderia interrogar a sua BC, para saber se há alunos que, sendo trabalhadores com mais de 50 anos, terminem o curso.

d) indique uma forma pragmática de mostrar, facilmente, que a BC em causa não está completa, isto é, não contém informação para todos os casos possíveis.

e) supondo que se pretendia armazenar, também, na BC informação sobre o universo que foi estudado —lista das universidades observadas, total de cursos oferecidos por cada, número total de inscritos e número de inscritos por grupo (alunos normais, trabalhadores e retardatários, período (ano inicial e ano final) de observação—, indique os predicados que acrescentaria para fazer esse registo.

Questão 2: Árvores de Prova e de Procura

Suponha que existe uma BC com 7 factos correspondentes ao predicado `dias/2` que associa a cada dia da semana o seu antecessor

```
dias(dom,sab).
dias(seg,dom).
dias(ter,seg).
dias(qua,ter).
dias(qui,qua).
dias(sex,qui).
dias(sab,sex).

hoje(seg).
```

e ainda um facto, `hoje/1`, que identifica o dia actual.
Responda, então, às alíneas seguintes:

a) escreva um predicado `acrescenta/0` que, a partir do dia de hoje, determina qual foi o dia de ontem e junta à BC um facto `ontem/1` com essa informação.

b) escreva um predicado `anteontem/2` que sucede (é verdadeiro) se o segundo argumento for o dia antes de ontem em relação ao primeiro argumento; p.ex:

```
?- anteontem(dom,sex).
   yes
?- anteontem(sex,AO).
   AO = qua
```

Admita, ainda, que a BC em causa também contém informação sobre os dias em que um indivíduo está ocupado a trabalhar e os dias em que uma pessoa comeu peixe, ou carne

```
comeupeixe(seg).
comeupeixe(qua).
comeupeixe(sex).
comeucarne(D) :- dias(D,_), not( comeupeixe(D) ).

ocupado(joao,dom).
ocupado(joao,ter).
ocupado(joao,qui).
ocupado(joao,sex).

invalido :- dias(H,O), verifica(H,O).
verifica(H,O) :- ocupado(P,H), ocupado(P,O),
                 write('Situacao impossivel').
```

e responda, agora, às alíneas seguintes:

c) usando uma *Árvore de Procura*, mostre qual seria a primeira resposta calculada pelo interpretador de Prolog à questão

```
?- comeucarne(D).
```

e indique, também, quais seriam as outras respostas calculadas a seguir, se se introduzisse '`;`' após cada resultado.

d) interprete o predicado `invalido/0` à luz da definição clausal apresentada acima.

e) usando uma *Árvore de Prova*, prove que o interpretador Prolog daria a seguinte resposta

```
?- invalido.
   yes
```

f) explique porque é que, embora sendo permitido, não teria qualquer efeito prático introduzir um predicado cut ('`!`') na cláusula que define `comeucarne/1`, como se mostra a seguir

```
comeucarne(D) :- dias(D,_), !, not( comeupeixe(D) ).
```

Questão 3: Manuseamento de Listas

Sobre operações com Listas em Prolog, responda às alíneas seguintes:

a) Pretende-se que implemente um predicado `iterador/3` que, dados um valor e uma lista, multiplique o valor por cada elemento da lista, formando uma outra lista com os produtos. Exemplo:

```
?- iterador(4,[1,2,3],L).
   L = [4,8,12]
```

b) Ainda em relação ao predicado anterior, pretende-se agora que o estenda para `iterador/4` de modo a receber como primeiro argumento a operação binária que se pretende aplicar ao valor (segundo argumento) e a cada elemento da lista. Exemplo:

```
?- iterador(adicao,4,[1,2,3],L).
   L = [5,6,7]
```

c) Pretende-se que implemente um predicado `empares/2` que, dada uma lista, forme uma outra lista de pares, em que cada elemento da primeira lista emparelha com o seu simétrico (o primeiro com o último, o segundo com o penúltimo, e assim sucessivamente). Exemplos:

```
?- empares([1,2,3],L).
   L = [(1,3),(2,2)]

?- empares([1,2,3,4],L).
   L = [(1,4),(2,3)]
```

d) Pretende-se que implemente um predicado denominado `une/3`, que serve para fundir duas listas ordenadas produzindo uma terceira lista, também ela ordenada, mas evitando repetidos. Exemplo:

```
?- une([2,5,6,8],[3,5,8,9],L).
   L = [2,3,5,6,8,9]
```

%———————————————————————————————————————
%———————————————————————————————————————

Questão 4: Árvores Binárias e Irregulares

Numa árvore binária todos os nodos, ou são vazios, ou então tem um valor e duas subárvores.
Para representar uma árvore irregular como se fosse binária, considera-se que a subárvore esquerda corresponde ao filho mais velho (mais à esquerda) e que a subárvore direita corresponde ao irmão que lhe fica imediatamente 'a direita.
Considere, então, a seguinte árvore genealógica (que é uma árvore irregular)

```
1: joao
 2: nuno
  3: ana
  3: rui
  3: sara
 2: marta
  3: jose
   4: olga
 2: maria
  3: pedro
  3: sandra
```

Responda, então, às alíneas seguintes:

a) escreva uma cláusula Prolog —correspondente ao facto `arv/3`— que descreva a árvore acima na forma de uma árvore binária (use a constante `vaz` para denotar a árvore vazia).

b) escreva uma outra cláusula alternativa à anterior, por exemplo designand-a agora por `arvgen/2`, em que descreva a mesma árvore genealógica mas usando a definição de árvore irregular (*um valor no nodo com zero ou mais subárvores*).

c) assumindo a representação que usou na alínea a), escreva um predicado `descendentes/2` que receba como primeiro argumento uma árvore e conte (devolvendo o resultado como segundo argumento) o total de descendentes do elemento na raiz.

d) assumindo a representação que usou na alínea b), escreva um predicado `repetidos/3` que receba como primeiro argumento um nome, como segundo uma árvore e devolva, como terceiro, uma lista contendo todas as ocorrências desse nome na árvore, associando a cada uma o nível em que aparece (a raiz é considerada como sendo o n'ıvel 1)

Questão 5: Processamento de Linguagens

Analise atentamente a seguinte gramática escrita na notação lógica DCG, cujo simbolo inicial é `concorrentes`

```
concorrentes -->  concorrente, outros.
outros       -->  [';'], concorrente, outros.
outros       -->  ['.'].
concorrente  -->  ident, classe.
ident        -->  nome, sexo, idade.
classe       -->  pontos.
nome         -->  ['Concorrente', id(N)].
sexo         -->  [masc].
sexo         -->  [fem].
idade        -->  [num(I), anos].
pontos       -->  ['Pontuac.', num(I)].
```

Responda, então, às alíneas seguintes:

a) Dê exemplo de 1 frase válida da linguagem definida pela DCG acima, desenhando a respectiva árvore de derivação (que do simbolo inicial lhe permite chegar aos terminais da sua frase).

b) Modifique a gramática supra de modo a permitir incluir uma lista de pontuações em cada concorrente.

c) Associe atributos aos símbolos e junte acções semânticas à DCG acima para: (1) a linguagem em causa só aceitar concorrentes que tenham idade entre os 15 e os 30 anos; (2) o programa reconhecedor (que se deriva da DCG) acrescentar à BC o predicado `pontuou/2` que tenha como argumentos o *nome* de cada concorrente e a respectiva *pontuação*.

d) Pretende-se, agora, que desenhe o autómato determinista necessário para reconhecer os símbolos terminais (as palavras reservadas e os nomes e números) da gramática acima apresentada.
Represente o autómato em causa numa BC em Prolog.