Paradigmas da Programação I
MiECom (2º ano)

Exame de Recurso

Data: 6 de Fevereiro de 2008
Hora: 09:30H – 12:00H

Questão 1: Representação de Conhecimento

Usando a abordagem seguida nas aulas para modelar sistemas de informação em lógica, construa uma Base de Conhecimento (BC) que descreva o SGCFP, um Sistema para Gestão de Candidaturas a Fundos Públicos.
Para isso sabe-se que uma Candidatura é descrita por um *código*, um *título*, uma *descrição* sumária e a *linha de acção* a que concorre e é proposta por uma ou mais Entidades (empresas públicas ou privadas) e por um ou mais Projectos. Cada Entidade é descrita por um *nome*, o *NIF*, a *morada*, o *regime de IVA* e o *nome do Responsável*. Quanto ao Projecto é descrito por um *resumo* uma *data de início* e uma *data de fim* e pelas Fases, que são caracterizadas pelo código da Entidade responsável (*NIF*) pela sua execução, pelo *titulo* pelas *datas de início e fim* de fase e pelo *custo*.
Modele o seu sistema de modo a que seja fácil: calcular o custo total de cada Projecto e da Candidatura global; calcular o número de fases por projecto ou de projectos por candidatura; e ainda validar que o período de cada fase esteja contido no período global do projecto – aponte a forma como o faria, essas operações.
Indique ainda algum tipo de perguntas que poderiam ser respondidas pela sua BC.

Questão 2: Bases de Conhecimento

Na BC do SGEEB, Sistema para Gestão do club/Escola de Equitação de Braga, existem entre outros os factos e as regras que se listam abaixo, só parcialmente instanciados.

```
% pessoa( bi, nome, sexo, anoNasc, morada, telef, email, habilitacoes).
% papel( biPessoa, cargo), em que cargo = admin, monitor, ajudante, limpeza, manutenção, socio, aluno.
papel( 1234567, socio).
papel( 11228899, aluno).
papel( 456889, monitor).
% cavalo( nome, raça, idade, biDono ).
cavalo( zorba, lusitano, 5, 1234567 ).
cavalo( seta, lusitano, 6, 1234567 ).
cavalo( melro, garrano, 10, 11228899 ).
% aula( biProf, biAluno, dia, hora, cavalo ).
$ equipamento( ref, designacao, qt ).
aloca( Prof, Aluno, D, H, Cavalo ) :- aula( Prof, Aluno, D, H, Cavalo ), !.
aloca( Prof, Aluno, D, H, Cavalo ) :- assertz(aula( Prof, Aluno, D, H, Cavalo )).
aloca1( Prof, Aluno, D, H ) :- aula( Prof, Aluno, D, H, _ ), !.
aloca1( Prof, Aluno, D, H ) :- escolhe( Aluno, Cavalo), valida( Cavalo, D, H ),
                               assertz(aula( Prof, Aluno, D, H, Cavalo )).
escolhe( Pess, Cav ) :- cavalo( Cav, _, _, Pess ).
escolhe(  _,   Cav ) :- cavalo( Cav, lusitano, _, _ ).
escolhe(  _,   Cav ) :- cavalo( Cav, garrano, I, _ ), I >=2, 10=<I.
valido(  Cav, D, H ) :- aula( _, _, D, H, Cav ), !, fail.
valido(  Cav, D, H ) .
```

Tomando em consideração a Base de Conhecimento (BC) constituída pelas regras e factos acima, responda às alíneas seguintes:

a) Porque razão designamos as *cláusulas de Horn* acima umas por *factos* e outras por *regras*?

b) Diga em que circunstâncias um Cavalo vai ser escolhido, segundo o critério acima expresso pelo predicado `escolhe/2`.

c) Explique por palavras suas a lógica do predicado `aloca/5`.

d) Explique em que difere realmente `aloca1/4` de `aloca/5`

e) Diga que questão devia colocar ao seu interpretador Prolog para saber todos os cavalos que pertencem a uma dada pessoa identificada pelo seu BI, obtendo o primeiro e forçando os restantes com o operador ";". Modifique a questão anterior, escrevendo um novo predicado que lhe liste automaticamente os cavalos do dono dado.

f) Escreva um predicado `questao1/2` para saber o nome e ano de nascimento de todos os alunos de sexo feminino ("f").

g) Escreva um predicado `questao2/3` para saber o BI, nome e telefone de todos os sócios que possuem um cavalo.

h) Observe as 2 seguintes questões

```
?- papel( BI, aluno).
?- papel( BI, Aluno).
```

e diga qual a diferença entre ambas; indique a resposta dada pelo Interpretador de Prolog a cada uma delas. Se a ordem dos 3 factos `papel` da BC fosse alterada, isso iria provocar alguma alteração na resposta calculada às 2 questões acima?

i) Escreva um predicado `limpa/0` que remove da BC todos os *cavalos árabes com mais de 20 anos.*

Questão 3: Manuseamento de Listas

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

a) Pretende-se que implemente um predicado `troca/2` que, dada uma lista de pares (a,b), devolva como segundo argumento a mesma lista com todos os pares trocados. Exemplo:

```
?- troca([(1,a),(2,b),(3c)], C).
   C = [(a,1),(b,2),(c,3)]
```

b) Pretende-se que implemente um predicado `somaPesa/3` que, dada uma constante C e uma lista de números, devolva como terceiro argumento o somatório dos elementos dessa mesma lista mas depois de cada um ser multiplicado pela constante C. Exemplo:

```
?- troca(2, [1,2,3], C).
   C = 12
```

c) Considere o predicado `p/3`, da forma `p(L1,L2,R)`, assim definido:

```
p( [], [], 0 ) :- !.
p( [H | T1], T2, X ) :- member(H,T1), !, p( T1, T2, Y), X is Y+1.
p( [H | T1], [H | T2], X ) :- p( T1, T2, X).
```

e explique por palavras suas o significado de `p/3`. Consolide a sua resposta, dizendo qual a resposta dada pela interpretador de Prolog à seguinte questão:

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

Questão 4–Autómatos Reactivos: Calculadora Simples

Pretende-se neste caso modelar o comportamento de uma *calculadora muito simples*, que efectua as 6 operações aritméticas elementares sobre números inteiros.
A *calculadora* aceita dígitos (de 0 a 9) que formam um *operando*, ou então um *operador* ($+$, $-$, $*$, $/$, $\uparrow$ (potência)) ou um sinal ($=$ ou C). A sequência começará, normalmente, por um *operando*, mas aceita também no início o *operador unário* $+$ ou o $-$. *Operadores* a seguir a um *operador* serão ignorados.
A *calculadora* é capaz de realizar várias operações, antes de apresentar o resultado, sem respeitar prioridades, isto é, realiza-as pela ordem em que aparecem. Após mostrar o resultado (quando lê o *sinal* $=$), apaga o resultado e volta ao início. O sinal C, lido após um digito ou um operador, tem como efeito apagar, apenas, esse último elemento.

Implemente, então, em Prolog esse **autómato determinista reactivo** (autómato determinista normal acrescido com acções semânticas nas transições), escrevendo apenas os predicados `automato(T,A,Q,S,Z)` e `delta(Q,T,Q1,A)`.