Primeira Prova de Técnicas Alternativas de Programação (07/10/2010)

Questão 1 (20 pontos)

Explique, com alguns detalhes, o que você entende pelos conceitos de Unificação e Backtracking em Prolog. Exemplifique sua apresentação.

Questão 2 (20 pontos)

Determine se o Prolog vai responder **yes** ou **no** como resultado das tentativas de unificação abaixo. Caso ele responda **yes**, dê o resultado da instanciação para cada variável.

```
?- media([32,33,34],M) = media([X,Y,Z],22).
?- f(X, a(b,X)) = f(Z, a(Z,c)).
?- X = [X].
?- pred([],[1]) = pred(X,[Y|X]).
?- suc(pred(suc(pred(1)))) = suc(pred(Z)).
```

Questão 3 (20 pontos)

O predicado "todos_apagados" deveria ser a relação de um ítem com duas listas onde a lista do terceiro termo contém todos os elementos da lista do segundo termo sem qualquer ocorrência do ítem do primeiro termo. No entanto, seu código foi escrito com erro. Veja o código errado em Prolog listado abaixo:

```
todos_apagados(_, [], []).
todos_apagados(X, [X|Cau1], Cau1).
todos_apagados(X, [Cab1|Cau1], [Cab1|Cau2]):-
   todos_apagados(X, Cau1, Cau2).
```

Com esse código errado, o comportamento de "todos_apagados" é o seguinte:

```
?- todos_apagados(a, [a,b,a,c], Resposta).
Resposta = [b,a,c] ?
yes
```

Faça a correção necessária no código de "todos_apagados" para que o comportamento dele passe a ser o seguinte:

```
?- todos_apagados(a, [a,b,a,c], Resposta).
Resposta = [b,c] ?
yes
```

Questão 4 (20 pontos)

Considere o domínio de problemas de transformações de estados constituído de uma base com 5 (cinco) lacunas, duas peças pretas e duas peças brancas (uma lacuna fica sempre vazia). Os estados abaixo são exemplos de configurações antes e depois de um movimento singular:

Estado antes

	p	b	p	b

Estado depois

b	p		p	b

Os únicos movimentos singulares possíveis são os seguintes:

- Deslizar uma peça para a lacuna vazia, se ela for adjacente;
- Saltar com uma peça por cima de apenas uma outra peça para ocupar a lacuna vazia.

Construir um predicado denominado **proximo** o qual representa uma relação binária entre um estado antes de uma alteração singular (primeiro termo) e um estado depois dela (segundo termo). O comportamento dele é o expresso abaixo:

```
?- proximo([b, b, v, p, p], Novo).
Novo = [b, b, p, v, p] ?;
Novo = [b, v, b, p, p] ?;
Novo = [b, b, p, p, v] ?;
Novo = [v, b, b, p, p] ?;
no
```

Questão 5 (20 pontos)

Construa um predicado em Prolog, denominado **maior_da_lista**, como uma relação binária capaz de determinar o maior elemento da lista do primeiro termo e instanciá-lo no segundo termo. O comportamento do predicado é expresso abaixo.

```
?- maior_da_lista([3, 6, 2, 8, 5, 1], X).
X = 8 ?
yes
```

No caso de uso de outro predicado auxiliar, o mesmo deve ser definido junto com a resposta desta questão.

BOM TRABALHO!