

Primeira Prova de Técnicas Alternativas de Programação (03/05/2007)

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)

Qual é o resultado das seguintes tentativas de unificação (determine se o Prolog vai responder **yes** ou **no**. Caso ele responda **yes**, dê o resultado da instanciação para cada variável):

```
?- 3+1 = 1+3.
?- f(X, a(b,X)) = f(Z, a(Z,c)).
?- [2,3,4|Z] = [2|[3,4,5,6]].
?- pred([], [1]) = pred(X, [Y|X]).
?- suc(pred(suc(pred(1)))) = suc(pred(Z)).
```

Questão 3 (20 pontos)

Construa um predicado recursivo em Prolog, denominado **todos_sao_diferentes**, que seja uma relação unária capaz de determinar se todos os elementos de um conjunto são diferentes entre si. O comportamento do predicado é expresso abaixo.

```
?- todos_sao_diferentes([1,2,3]).
yes

?- todos_sao_diferentes([1,2,1]).
no
```

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

Questão 4 (20 pontos)

A conjectura de Goldbach diz que todo número par positivo maior que 2 (dois) pode ser obtido pela soma de 2 (dois) números primos (e.g., $28 = 5 + 23$). Construa um predicado em Prolog, denominado **soma_de_2_primos**, o qual expressa uma relação binária sobre um número inteiro e uma lista de exatamente dois números inteiros. Seu comportamento é o expresso abaixo.

```
?- soma_de_2_primos(28, L).
L = [5,23] ?
yes
```

Para facilitar a solução, assumo a existência de dois predicados, **soma_de_2_naturais** e **e_primo**, e use-os obrigatoriamente na definição do predicado **soma_de_2_primos**.

Os comportamentos do predicado **soma_de_2_naturais** são os expressos abaixo.

```
?- soma_de_2_naturais(56, [25,31]).
yes
```

```
?- soma_de_2_naturais(8, L).
L = [1,7] ? ;
L = [2,6] ? ;
L = [3,5] ? ;
no
```

Os comportamentos do predicado **e_primo** são os expressos abaixo.

```
?- e_primo(11).
yes
```

```
?- e_primo(P).
P = 1 ? ;
P = 2 ? ;
P = 3 ? ;
P = 5 ? ;
P = 7 ? ;
P = 11 ? ;
...
...
```

DICA: note bem que ambos os predicados **soma_de_2_naturais** e **e_primo** podem ser ativados com seus termos instanciados ou não pois permitem retroação (*backtracking*) inter-cláusulas.

Questão 5 (20 pontos)

São dados os fatos abaixo sobre o alfabeto:

```
sucessor(a,b).
sucessor(b,c).
sucessor(c,d).
...
sucessor(y,z).
sucessor(z,a).
```

Escreva um predicado ternário em Prolog denominado **codificada**, o qual é capaz de trocar cada letra de uma palavra por uma outra letra, a qual é calculada a partir de n deslocamentos do **sucessor** da letra a partir de sua posição original no alfabeto. Por exemplo, se $n = 3$, a letra d será substituída pela letra g , e a letra z pela letra c . O predicado relaciona uma lista de letras que representa a palavra original (primeiro termo) com o deslocamento n (segundo termo) e instancia uma lista de letras que representa a palavra codificada (terceiro termo). Um exemplo de ativação do predicado **codificada** é o seguinte:

```
?- codificada([s,a,l,a,d,a], 4, X).
X = [x, e, p, e, h, e] ?
yes
```

BOM TRABALHO!