

Recursividade em Prolog
Lista de Exercícios
Laboratório (31 de Maio de 2009)
Revisão – Aquecimento

Usando o conceito de recursividade, faça os predicados/regras solicitados¹:

1. Antes de começar revise e execute os exemplos da apostila. Feito isto, parabéns, então siga frente.
2. Traduza e ilustre com exemplos, uma solução recursiva para os textos abaixo:
 - (a) Seja o conceito de ancestral: "*If Y is a parent of X, then Y is an ancestor of X. if Y is an ancestor of Z and Z is a parent of X, then Y is an ancestor of X*". Reuse esta solução, para gerar um conceito de descendente.
 - (b) *Voce fala bem alguém, se voce conhece este alguém. Ou ainda, voce fala bem sobre uma pessoa Y (alguém), se voce conhece uma pessoa Z e Z a qual fala bem sobre a pessoa Y. Inspirado em: You talk about someone either if you know them or you know someone who talks about them.*
 - (c) *Seja o conceito de número natural dado pelo seguintes regras recursivas: "0 is represented as 0 N + 1 is represented as s(X), where X is a representation of N". Demonstre que 7 é um número natural.*
3. Gerar e imprimir uma seqüência de números em ordem **crescente**, do tipo: 1 2 3 4 ... 99. Protótipo da regra: gera_cresc(N).
4. Uma variante da anterior, é entre dois números quaisquer $N1$ e $N2$, tal que $N2 > N1$. Logo um protótipo da regra é do tipo: gera_cresc($N1, N2$).
5. Gerar a soma dos múltiplos de **7** até um valor N (verificar se este também é múltiplo de 7, ou maior até o último múltiplo) a partir de 0 ou 7.
6. A soma recursiva entre dois números $N1$ e $N2$, tal que $N2 > N1$.
7. Construir um conjunto de regras tal que imprima uma certa quantidade N de * (asteriscos) em uma linha corrente. Protótipo da regra: ast(N).
8. Aproveitando as regras do item anterior, imprimir estes asteriscos de modo a saída forme um triângulo do tipo:

```
*****
****
***
**
*
```

A idéia é que a cada linha, o número de * a serem impressos diminua. Protótipo da regra: triang(N).

9. Um conjunto de triângulos tipo fractal Dificuldade razoável e apenas use a idéia do anterior.
10. Imprimir asteriscos, de modo que a saída sejam quadrados. Ex: ?- quad(3), gera uma saída do tipo:

```
***
***
***
```

Isto é, um quadrado formado por 3 linhas, tem 3 asteriscos em cada uma. Protótipo da regra: quad(N).

Dica inicial: quad(N):- quad_aux(Linhas,Colunas).

11. Predicados que calculem a função de Fibonacci onde $fib(x) = 1$ se $x \leq 2$ e, caso contrário, $fib(x) = fib(x - 1) + fib(x - 2)$. Está no texto.
12. Um conjunto de regras que encontre o Máximo Divisor Comum (MDC) entre dois números X e Y . Protótipo da regra: mdc(X,Y,Max_div). Pesquise pelo Algoritmo de Euclides.
13. Um predicado que identifique se um número é primo ou não.
14. Reusando o anterior, imprima uma sequência de números primos.
15. Para um n par, calcule a soma de uma série matemática tipo:

$$\frac{2}{3} + \frac{4}{5} + \frac{6}{7} + \dots + \frac{n}{n+2}$$

16. Construa um conjunto de regras que faça um sistema de menu (ver apostila), mas que cada uma das ações, seja algum dos problemas acima. Ou seja, incorporar alguns problemas sob um sistema de menu.
17. Pesquise e resolva o problema da torre de Hanoi.

Como 17 é um número primo, ficamos por aqui!

¹Para tornar efetivo o aprendizado, ative o *guitracer* e execute com o *trace* ativado em alguns casos.

*** **
** **
* *
*** **
** **
* *

**
*