GSI010 - Programação Lógica

Revisão aula 3

- função recursiva: função que faz chamada a ela mesma
- técnica de programação
- divisão de um problema grande em pedaços menores
- ▶ algumas linguagens são puramente recursivas: Prolog e Lisp

Problemas:

- fatorial: n! = n * (n-1)!, 0! = 1
- fibonacci: $f_n = f_{n-1} + f_{n-2}$, $f_1 = 1$, $f_2 = 2$
- estruturas de dados em árvores
- grafos, redes sociais

Recursão

Estratégia de resolução:

- 1. Definir caso base
 - ▶ fatorial:0! = 1
 - fibonacci: $f_1 = 1$, $f_2 = 2$
 - ▶ em árvores: nó-folha
- 2. Definir operação para subproblema base
- 3. Combinar resultado de subproblema com parte maior

Exemplo: versão recursiva

```
filho (ana, brigite).
filho (brigite, caroline).
filho (caroline, donna).
filho (donna, emilia).
```

Exemplo: versão recursiva

```
filho(ana, brigite).
filho(brigite, caroline).
filho(caroline, donna).
filho(donna, emilia).

descende(X, Y):- filho(X,Y).
descende(X, Y):- filho(Z, Y), descende(X, Z).
```

Exemplo: versão recursiva

```
filho (ana, brigite).
filho (brigite, caroline).
filho (caroline, donna).
filho (donna, emilia).

descende(X, Y):- filho (X,Y).
descende(X, Y):- filho (Z, Y), descende(X, Z).
```

```
?- descende(ana, donna).

true;
false.
```

Esta aula: aritmética

Operadores aritméticos

- ▶ As operações +, , *, / são descritas por predicados especiais
- +/2, -/2, */2, (/)/2 são de aridade 2
- ► +, -, *,/ são functors em Prolog
- ▶ is é obrigatório para interpretar esses operadores

```
1 somaTresDepoisDuplica(X, Y):-
2 Y is (X+3)*2.
```

O operador **is**

Condições especiais de uso

variáveis tem que estar previamente unificadas

Exemplo de uso em variável não unificada:

```
1 ?- X is Z+2.
2 ERROR: is /2: Arguments are not sufficiently instantiated
```

Operadores lógicos

- **▶** 5 > 3.
- ▶ maior ou igual: 5 >= 123.
- ▶ estritamente lógico: 4=:=3.
- ▶ diferente: 5 = 2.

Unificação vs. igual lógico

Não confundir

- = com == com =:=
 - ► = é unificação
 - ► == verifica se valores são iguais
 - ► =:= funciona somente com números

Exercício

Fazer predicado fibonacci(N, F). que calcula termo da sequência de Fibonnaci.

Regras de recursão

$$f_2 = 2$$

 $f_n = f_{n-1} + f_{n-2}$

 $f_1 = 1$

```
fibonacci(1,1).
fibonacci(2,2).
fibonacci(N, Y) :- ...
```

Referências

- ▶ Luis, A. M. Palazzo, Introdução à programação prolog, Educat, 1997
- Slides profs. Elaine Faria, Hiran Nonato e Gabriel Coutinho -UFU
- ► Slides da Profa. Solange ICMC USP