

Programação em Lógica

Prolog, indução e recursão.

Prolog

- ♦ Fatos: definição dos elementos que pertencem à relação representada por um predicado.
 - `pai(carlos,maria), casados(carlos,regina), aula(65099,semestre(2002,2),7395PB,63).`
- ♦ Regras: derivação de novos fatos a partir de fatos e relações conhecidos.
 - `professor(X,Y):- disciplina(Z), matriculado(Y,Z), ministrante(X,Z).`

Prolog

- ♦ Listas:
 - `[], [1,2,3], [maria, 25, unisinos], ...`
 - estrutura geral de uma lista: `[X|Y]`, onde X é o primeiro elemento da lista e Y é a lista original sem o primeiro elemento. Exemplo: `[1,2,3]` unifica com `[X|Y]`, com `X=1` e `Y=[2,3]`.
 - Operações sobre listas: construídas a partir de regras.

Operações lógicas

- ♦ `,` — “e” lógico.
- ♦ `;` — “ou” lógico.
- ♦ `<, >, =<, >=` — significados usuais.
- ♦ `=, \=` — `X=Y` se X é (ou não é) unificável com Y.
- ♦ `=:, \:=` — `X=:Y` se X e Y são (ou não são) idênticos (representam exatamente o mesmo elemento instanciado ou não).
- ♦ `:=:, :=\=` — igualdade e desigualdade aritmética.

Uma questão de fé... ☺

- ♦ Indução matemática:
 - **base de indução:** prova-se que determinada propriedade é válida quando o tamanho do problema é igual a 0.
 - **hipótese de indução:** supõe-se que para um problema de tamanho menor ou igual a n, a propriedade é válida.
 - **passo de indução:** prova-se que a propriedade é válida para um problema de tamanho n+1.

Exemplo de prova por indução

- ♦ Seja Σ um alfabeto (conjunto finito de símbolos) e Σ^* o conjunto de todas as palavras que resultam na concatenação sucessiva de 0 ou mais símbolos de Σ .
 - Exemplo, se Σ é o alfabeto binário $\{0,1\}$, então $\Sigma^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, \dots\}$, onde ϵ representa a palavra vazia, ou a concatenação de 0 símbolos de Σ .
- ♦ Dada uma palavra $w \in \Sigma^*$, define-se a palavra reversa de w , w^R (ou w lida de trás para a frente) como sendo:
 - ϵ , se $w = \epsilon$
 - $u^R a$, se $w = au$, $a \in \Sigma$, $u \in \Sigma^*$
- ♦ Vamos então provar que: *Para quaisquer duas palavras $u, v \in \Sigma^*$, $(u \cdot v)^R = v^R \cdot u^R$.*

Prova por indução

- ♦ base de indução: $|u \cdot v| = 0$.
 - $(\varepsilon \cdot \varepsilon)^R = (\varepsilon)^R = \varepsilon = \varepsilon^R = \varepsilon^R \cdot \varepsilon = \varepsilon^R \cdot \varepsilon^R$
- ♦ hipótese de indução: sempre que $|u \cdot v| \leq n$, $(u \cdot v)^R = v^R \cdot u^R$.
- ♦ passo de indução: $|u \cdot v| = n + 1$.
 - $(ax \cdot v)^R = [u = ax, \text{ já que } u \text{ deve ter pelo menos um símbolo}]$
 - $(a \cdot (x \cdot v))^R = [a \text{ concatenação de palavras é associativa}]$
 - $(x \cdot v)^R \cdot a = [\text{pela definição de palavra reversa}]$
 - $v^R \cdot x^R \cdot a = [\text{pela hipótese de indução: } x \cdot v \text{ tem } n \text{ símbolos}]$
 - $v^R \cdot (ax)^R = [\text{pela definição de palavra reversa}]$
 - $v^R \cdot u^R [u = ax], \text{ como queríamos demonstrar.}$

A mesma questão de fé...

- ♦ Programação recursiva:
 - **base de recursão:** soluciona-se algoritmicamente o problema para quando ele tem tamanho 0 .
 - **chamada recursiva:** supõe-se que o algoritmo funciona para o problema quando esse tem tamanho $n-1$.
 - **algoritmo recursivo:** soluciona-se algoritmicamente o problema quando este tem tamanho n .

Exemplo de programa recursivo

```
lista_reversa([], []).
lista_reversa([X|Y], R) :-
    lista_reversa(Y, L), concatenar(L, [X], R).

base de recursão
chamada recursiva
algoritmo recursivo
```

Exercícios

- ♦ Dada uma lista L, implementar um predicado ordenar(L, LOrdenada), onde LOrdenada é a lista L com os seus elementos ordenados.
- ♦ Implementar a noção de conjuntos (finitos) e as operações sobre conjuntos $\in, \notin, \subseteq, \supseteq, \cup, \cap$.
- ♦ Implementar a noção de árvore binária, e construir os predicados que verificam se uma determinada árvore é uma árvore binária de pesquisa, se um determinado elemento pertence a uma árvore binária de pesquisa e se duas árvores são isomorfas. Implementar também os diferentes caminhamentos em árvore.