

Programação Lógica

Prolog: predicados de entrada e saída, de negação, de mesmo nome, interpretados, compilados, estáticos e dinâmicos

Profa Heloisa de Arruda Camargo

Priscilla de Abreu Lopes (PESCD)
priscilla_lopes@dc.ufscar.br

Entrada e Saída

- Predicado **read**
- Sintaxe: `read(Termo)`
onde ?Termo (variável ou átomo)
- Lê um termo do dispositivo de entrada corrente e unifica com Termo. O termo dado deve ser seguido de . (ponto)

Entrada e Saída

- Predicado **write**
- Sintaxe: `write(Termo)`
onde ?Termo (termo)
- Escreve o termo no dispositivo de saída corrente
- Predicado **nl** - muda para próxima linha no dispositivo de saída

Entrada e Saída

```
| ?- read(X), Y is X + 1.  
|: 3.  
X = 3 ,  
Y = 4
```

Negação

- Predicado **not** (operador lógico de negação)
- Sintaxe: not G
onde +G (G é um objetivo)
- Retorna sucesso se G for falso e vice-versa
- Retorna **erro** se alguma variável em G estiver livre e for instanciada como consequência da execução de G.

Negação

```
peessoa(rosa).
peessoa(carlos).
peessoa(fabio).
|?- peessoa(maria).
no
|?- not(peessoa(maria)).
yes
|?- not(peessoa(carlos)).
no
|?- not(peessoa(Alguem)).
ERRO
```

Negação

- Predicado **\+** (negação não lógica)
- Sintaxe: \+ G
onde +G (G é um objetivo)
- Retorna sucesso se G for falso e vice-versa
|?- \+ peessoa(Alguem).
no
|?- \+(pertence(X,[1,2,3]),pertence(X,[4,5,6])).
yes

Predicados de Mesmo Nome

- Um predicado é identificado pelo seu nome e pela aridade (número de argumentos).
- Predicados com o mesmo nome e com número de argumentos diferentes são considerados **diferentes**.
- Os predicados sem argumentos são normalmente usados para identificar procedimentos que usam **read** e **write**.

Predicados de Mesmo Nome

- Soma dos elementos de uma lista numérica

```
soma :- write('Digite uma lista de numeros'),
        read(Lista),
        soma(Lista,Resultado),
        write('A soma dos elementos da lista e = '),
        write(Resultado),
        nl.
soma([ ],0).
soma([Elem|Cauda], S) :- soma(Cauda,S1),
                        S is S1 + Elem.

| ?- soma.
Digite uma lista de numeros: [4,5,6,4,4,0.3,-7].
A soma dos elementos da lista e 12.7
yes
```

Predicados Interpretados, Compilados, Estáticos e Dinâmicos

- Predicados em Prolog podem ser interpretados ou compilados
 - Código compilado:
 - estático
 - não pode ser listado
 - roda mais rápido
 - Código interpretado:
 - estático ou dinâmico
 - pode ser listado
 - roda mais lentamente

Predicados Interpretados, Compilados, Estáticos e Dinâmicos

- Alteração de procedimentos dinâmicos:
 - Adicionar cláusulas(assert)
 - Eliminar cláusulas(retract)
 - Recuperar (clause)
- Procedimentos definidos com consult ou compile são estáticos por default
- Para serem modificados, devem ser declarados como dinâmicos

Predicados Interpretados, Compilados, Estáticos e Dinâmicos

- A definição do procedimento deve ser precedida por
:- dynamic gosta/2.

Exercício

- Escreva um programa Prolog para dado um elemento, uma lista e um número inteiro que indica uma posição nessa lista, inserir o elemento na posição indicada. Se a posição indicada for maior que o tamanho da lista, inserir no final da lista.