Programação Lógica

Prolog: operadores aritméticos e relacionais, comparação entre termos, exemplos

Profa Heloisa de Arruda Camargo

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

Aritmética em Prolog

- Operadores aritméticos são funtores
  - 2+5 é representado internamente como **+(2,5)**
  - Ativação das funções por meio do predicado **IS**:
    - X is <expressão aritmética>, onde X é uma variável
  - Calcula a expressão e instancia a variável X com o resultado

Aritmética em Prolog

? - X is 1+3.

X = 4

? - X is 4*3+10/2.

X = 17

? - X is abs((15-30)//2).

X = 7

Aritmética em Prolog

- Outros operadores:
  - X+Y
  - X-Y
  - X*Y
  - X/Y
  - X//Y (divisão inteira)
  - X^Y (exponenciação)
  - -X
  - X mod Y

Aritmética em Prolog

- Outros operadores:
  - abs(X)
  - exp(X)
  - ln(X)
  - log(X)
  - sin(X)
  - cos(X)
  - sqrt(X)

Operadores Relacionais

- E1 e E2 são expressões aritméricas, calculadas antes da aplicação do operador
  - E1 < E2
  - E1 > E2
  - E1 <= E2
  - E1 >= E2

Operadores Relacionais

- E1 e E2 são expressões aritméricas, calculadas antes da aplicação do operador
  - X is E1 %calcula E1 e unifica com X
  - E1 =:= E2 %igualdade
  - E1 =/= E2 % desigualdade

Operadores Relacionais

```
? - 2+1 < 6-2.
true
? - 7-3 > 4+2.
fail
? - 1+2 =:= 2+1
true
? - 1+2 =:= X.
ERROR: =:=/2: Arguments are not sufficiently
    instantiated
```

Comparação entre Termos

- Unificação de termos: predicado **=**
  - Sintaxe: Termo1 = Termo2
  - Retorna sucesso se os termos Termo1 e Termo2 unificam:

    ?- 2 = 2.

    true.

    ?- maria = maria.

    true.

    ?- X = Y.

    X = Y.

Comparação entre Termos

?- X = 2+5.

X = 2+5.

?- X is 2+5.

X = 7.

?- X = 2+5, Y is X.

X = 2+5,

Y = 7.

Comparação entre Termos

?- 1+2 = 2+1.

fail.

?- 1+2 =:= 2+1.

true.

Comparação entre Termos

- Verificação de termos idênticos: predicado **==**
  - Sintaxe: Termo1 == Termo2
  - Retorna sucesso se os termos Termo1 e Termo2 são idênticos:

    ?- nome == nome.

    true.

    ?- X == X.

    true.

Comparação entre Termos

?- X == 5.
fail.

?- X = 5.
X = 5.

?- X == Y.
fail.

Comparação entre Termos

?- not(1) == not(X).
fail.

?- not(1) = not(X).
X = 1.

Comparação entre Termos

?- X = 1+2.
X = 1+2.
?- X is 1+2.
X = 3.
?- X == 1+2.
fail.
?- X =:= 1+2.
ERROR: =:=/2: Arguments are not sufficiently instantiated

Comparação entre Termos

- Verificação de termos não idênticos: predicado **\==**
  - Sintaxe: Termo1 \== Termo2
  - Retorna sucesso se os termos Termo1 e Termo2 **não** são idênticos:
    ?- nome \== nome.
    fail.
    ?- X \== Y.
    true.

Comparação entre Termos

- Outras comparações:
  - Sintaxe: Termo1 <op> Termo2
  - <op> pode ser: @>=, @>, @=<, @<
  - Compara os termos sem calcular:

```
?- 2+1 < 1+3.
true.
?- 2+1 @< 1+3.
fail.
?- 2 @< 1+3.
true.
```


Exemplos

- SWI-Prolog
  - http://www.swi-prolog.org/
  - Linux, Windows, MacOSX
- Amzi! Prolog
  - http://www.amzi.com/index.html

Exemplos

- Somar os elementos de uma lista numérica

```
soma([ ],0).
soma([Elem| Cauda], S) :-
    soma (Cauda,S1),
    S is S1 + Elem.

?- soma([1,2,3,4,5,6], S).
S = 21
```


Exemplos

- Contar o número de elementos de uma lista

```
conta([ ],0).
conta([_| Cauda], N) :-
    conta(Cauda, N1),
    N is N1 + 1.

?- conta([1,2,3,4,5,6],C).
C = 6
```

Exemplos

- Eliminar todas as ocorrências de um elemento de uma lista

```
del_todas(Elem,[ ],[ ]).
del_todas(Elem, [Elem|Y], Z) :-
    del_todas(Elem,Y,Z).
del_todas(Elem,[Elem1|Y], [Elem1|Z]) :-
    Elem \== Elem1,
    del_todas(Elem,Y,Z).
```


Exemplos

?- del_todas(a, [a,b,a,c,d,a,e], L).
L = [b, c, d, e]

?- del_todas(a, [a,b,a,d,[a], e], L).
L = [b, d, [a], e]

Exemplos

- Retirar todas as repetições de uma lista

```
retirar_rep([ ],[ ]).
retirar_rep([Elem|Cauda],[Elem|Cauda1]) :-
    del_todas(Elem,Cauda,Lista),
    retirar_rep(Lista,Cauda1).
```

```
?-retirar_rep([a,b,[a],c,b,x,p1,b,a,[a]],Resultado).
Resultado=[a,b,[a],c,x,p1].
```


Exemplos

- Contar o número de ocorrências de um dado elemento no primeiro nível de uma lista

```
conta_ocorr(Elem,[ ],0).
conta_ocorr(Elem,[Elem|Y],N) :-
    conta_ocorr(Elem,Y,N1),
    N is N1 + 1.
conta_ocorr(Elem,[Elem1|Y], N) :-
    Elem \== Elem1,
    conta_ocorr(Elem,Y,N).
```

Exemplos

?- conta_ocorr(1,[1,2,3,1,4,5,1,6,7,1],N).
N = 4 .

?- conta_ocorr(1,[1,2,3,[1,4,5],1,6,7,[1]],N).
N = 2 .

Exercício

- Dada uma lista de números, separar em duas sendo uma com os positivos e outra com os negativos, descartando o zero

?- separa_sz([5,4,3,2,9,-8,-10,0,2,34,56,-77],P, N).
P = [5, 4, 3, 2, 9, 2, 34, 56],
N = [-8, -10, -77] .