

Paradigmas de Programação

Prof. Maicon R. Zatelli

Prolog - Programação Lógica
Estruturas

Universidade Federal de Santa Catarina
Florianópolis - Brasil

Prolog

Representando uma árvore binária.

```
no(52,  
    no(32,  
        no(12,nil,nil),  
        no(35,nil,nil)),  
    no(56,  
        no(55,  
            no(55,nil,nil),  
            nil),  
        no(64,nil,nil)))
```

Prolog

```
soma(nil, 0).  
soma(no(N, ArvE, ArvD), Soma) :-  
    soma(ArvE, SomaEsq),  
    soma(ArvD, SomaDir),  
    Soma is N + SomaEsq + SomaDir.
```

- Se a árvore é nula, então a soma dos nós é 0.
- Caso contrário, ela é a soma de N com a soma da árvore esquerda e da árvore direita.

Prolog

```
?- no(X,E,D), soma(no(X,E,D),Soma).
```

Prolog

```
?- no(X,E,D), soma(no(X,E,D),Soma).  
X = 52,  
E = no(32, no(12, nil, nil), no(35, nil, nil)),  
D = no(56, no(55, no(55, nil, nil), nil), no(64, nil, nil)),  
Soma = 361.
```

Prolog

```
membro(N, no(N, _, _)) :- !.  
membro(N, no(_, ArvE, ArvD)) :-  
    membro(N, ArvE), !;  
    membro(N, ArvD).
```

- N é membro da árvore se ele é igual ao elemento da raiz da árvore.
- Caso contrário, ele será membro da árvore se ele for membro da árvore esquerda ou for membro da árvore direita.

Prolog

```
?- no(X,E,D), membro(55,no(X,E,D)).
```

Prolog

```
?- no(X,E,D), membro(55,no(X,E,D)).  
X = 52,  
E = no(32, no(12, nil, nil), no(35, nil, nil)),  
D = no(56, no(55, no(55, nil, nil), nil), no(64, nil, nil)).
```


Prolog

```
min(X,Y,X) :- X <= Y, !.  
min(X,Y,Y).  
  
minimo(nil,1000). %INF = 1000  
minimo(no(N, ArvE, ArvD),Min) :-  
    minimo(ArvE, MinE),  
    minimo(ArvD, MinD),  
    min(MinE, MinD, MinED),  
    min(N, MinED, Min).
```

- O mínimo de uma árvore nula é INF.
- Caso contrário, será o mínimo entre o mínimo da árvore esquerda, o mínimo da árvore direita e o elemento do nó.

Prolog

```
?- no(X,E,D), minimo(no(X,E,D),Min).
```

Prolog

```
?- no(X,E,D), minimo(no(X,E,D),Min).  
X = 52,  
E = no(32, no(12, nil, nil), no(35, nil, nil)),  
D = no(56, no(55, no(55, nil, nil), nil), no(64, nil, nil)),  
Min = 12.
```

Prolog

```
caminho(Y, no(Y, _, _), [Y]) :- !.  
caminho(Y, no(K, ArvE, ArvD), [K|Caminho]) :-  
    caminho(Y, ArvE, Caminho), !;  
    caminho(Y, ArvD, Caminho).
```

- Y faz parte do caminho da raiz até ele, se o nó atual contém Y.
- O nó K faz parte do caminho da raiz até Y, se há um caminho a partir dele até Y indo pela árvore esquerda ou indo pela árvore direita.

Prolog

```
?- no(X,E,D), caminho(55,no(X,E,D),Caminho).
```

Prolog

```
?- no(X,E,D), caminho(55,no(X,E,D),Caminho).  
X = 52,  
E = no(32, no(12, nil, nil), no(35, nil, nil)),  
D = no(56, no(55, no(55, nil, nil), nil), no(64, nil, nil)),  
Caminho = [52, 56, 55].
```

Prolog - Alguns Links Úteis

- `http://www.swi-prolog.org/pldoc/man?section=lists`
- `http://lpn.swi-prolog.org/lpnpage.php?pageid=online`
- `http://www.swi-prolog.org/pldoc/doc/_CWD_/index.html`

Prolog

Ver atividade no Moodle