

Processamento de Listas

- **Listas em Prolog – notação por ponto:**
 - “nil”, constante denotando a lista vazia
 - “.”, símbolo funcional binário, usado na notação infixa
 - se u é um termo Prolog e v é uma lista de termos Prolog, então (u.v) é uma lista de termos Prolog
- **Listas em Prolog – notação por colchete:**
 - [], constante denotando a lista vazia
 - “|”, símbolo funcional binário, usado na notação infixa
 - se u é um termo Prolog e v é uma lista de termos Prolog, então [u|v] é uma lista de termos Prolog

Prolog - Listas

- **Notação por ponto:**

- plena: $(e_1.(e_2.(...(e_n.nil)...)))$
- simplificada: $(e_1.e_2. \dots e_n.nil)$

- **Notação por colchete:**

- plena: $[e_1 | [e_2 | [...[e_n | []]...]]]$
- simplificada: $[e_1, e_2, \dots, e_n]$

Prolog - Listas

- **Exemplos:**

Notação por ponto

`nil`

`(a. b. c. d. nil)`

`((a. b. nil). c. (d. e. f. nil). g. nil)`

`((A11. A12. nil). (A21. A22. nil). nil)`

Notação por colchete

`[]`

`[a,b,c,d]`

`[[a,b],c,[d,e,f],g]`

`[[A11,A12],[A21,A22]]`

Prolog - Listas

- Exemplos:

lista: (a. b. c. d. nil)
 cabeça: a
 cauda: (b. c. d. nil)

lista: ((a. b. nil). c. (d. e. f. nil). g. nil)
 cabeça: (a. b. nil)
 cauda: (c. (d. e. f. nil). g. nil)

lista: ((A11. A12. nil). (A21. A22. nil). nil)
 cabeça: (A11. A12. nil)
 cauda: ((A21. A22. nil). nil)

lista: [a, b, c, d]
 cabeça: a
 cauda: [b, c, d]

lista: [[a, b], c, [d, e, f], g]
 cabeça: [a, b]
 cauda: [c, [d, e, f], g]

lista: [[A11, A12], [A21, A22]]
 cabeça: [A11, A12]
 cauda: [[A21, A22]]

Prolog - Listas

- Unificação de listas:

lista L_1 : (a. b. c. d. nil)

lista L_2 : (E1. E2. E)

unificação: E1/a
E2/b
E/(c. d. nil)

lista L_1 : ((a. b. nil). c. (d. e. f. nil). g. nil)

lista L_2 : (H. T)

unificação: H/(a. b. nil)
T/(c. (d. e. f. nil). g. nil)

lista L_1 : ((a. b. nil). c. (d. e. f. nil). g. nil)

lista L_2 : ((X. Y). Z)

unificação: X/a
Y/(b. nil)
Z/(c. (d. e. f. nil). g. nil)

lista L_1 : [a, b, c, d]

lista L_2 : [E1, E2 | E]

unificação: E1/a
E2/b
E/[c, d]

lista L_1 : [[a, b], c, [d, e, f], g]

lista L_2 : [H | T]

unificação: H/[a, b]
T/[c, [d, e, f], g]

lista L_1 : [[a, b], c, [d, e, f], g]

lista L_2 : [[X | Y] | Z]

unificação: X/a
Y/[b]
Z/[c, [d, e, f], g]

Prolog - Listas

1) **adiciona(X,L1,L2)** – onde L2 é a lista que contém o elemento X e a lista L1.

Testar este predicado no interpretador Prolog, executando:

?- adiciona(1,[2,3],L).

?- adiciona(X,[2,3],[1,2,3]).

```
adiciona(X,L,[X|L]).
```

Listas - Operações

apaga(X,L1,L2) – onde L2 é a lista L1 sem o elemento X. Testar com:

?- apaga(a,[a,b,a,c],L).

?- apaga(a,L,[b,c]).

```
apaga(X,[X|R],R).
```

```
apaga(X,[Y|R1],[Y|R2]):-  
    apaga(X,R1,R2).
```

Listas - Operações

membro(X,L) – que é verdadeiro se X pertencer à lista L. Testar com:
?- membro(b,[a,b,c]).

```
membro( X, [X|_] ).  
membro( X, [_|R] ) :- membro( X, R ).
```

Listas - Operações

concatena(L1,L2,L3) – onde L3 é resultado da junção das listas L2 e L1.

Testar com:

?- concatena([1,2],[3,4],L).

?- concatena([1,2],L,[1,2,3,4]).

?- concatena(L,[3,4],[1,2,3,4]).

```
concatena([],L,L),
```

```
concatena([X|L1],L2,[X|L3]):- concatena(L1,L2,L3),
```

Listas - Operações

`comprimento(X,L)` – onde X é o número de elementos da lista L . Testar com:
?- `comprimento(X,[a,b,c])`.

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

Listas - Operações

Há ocasiões em que, por um ou outro motivo, não desejamos a continuidade da verificação das regras. Para estes casos usamos um mecanismo para corte do fluxo, simbolizado por um ponto de exclamação (!).

O corte é útil quando sabemos que uma determinada regra é a última a ser analisada e haveria problemas na continuidade de verificação das demais regras. Por exemplo, uma implementação de função fatorial:

fat(0,1) :- !.

fat(N,R) :- Ni is N-1, fat(Ni,Ri), R is Ri*N.

Corte de fluxo

- O modelo relacional de banco de dados (BD)
 - Representa os dados em um BD usando um conjunto de **tabelas**.
 - As linhas das tabelas são denominadas **tuplas** e suas colunas, **atributos**.
 - As **consultas** são representadas em termos de operações da álgebra relacional (projeção, seleção, etc.).
- Programação em lógica é uma poderosa extensão do modelo relacional.
 - Um conjunto de fatos para um predicado corresponde a uma tabela.
 - Fatos representam tuplas e seus argumentos representam atributos da tabela.
 - Uma regra corresponde a uma vista no modelo relacional, isto é, uma tabela virtual que reúne atributos já armazenados em outras tabelas.
 - As operações de álgebra relacional podem ser representadas por meio de consultas (ou regras).

Modelo Relacional e Prolog

Primeira Forma Normal (1FN)

Todos os atributos devem ser atômicos.

Dados

Código	Nome	Salário	Dependentes
107	Ana	5.500,00	Bia, Lia
290	Rui	7.850,00	Raí
368	Eva	2.390,00	---
405	Ivo	4.700,00	Clô, Ary, Noé

Funcionários

Código	Nome	Salário
107	Ana	5.500,00
290	Rui	7.850,00
368	Eva	2.390,00
405	Ivo	4.700,00

Dependentes

Código	Nome
107	Bia
107	Lia
290	Raí
405	Clô
405	Ary
405	Noé

Dados na 1FN

Modelo Relacional e Prolog

Tabelas representadas em Prolog

```
% func(Código, Nome, Salário)
```

```
func(107, ana, 5500).
```

```
func(290, rui, 7850).
```

```
func(368, eva, 2390).
```

```
func(405, ivo, 4700).
```

```
% dep(Código, Nome)
```

```
dep(107, bia).
```

```
dep(107, lia).
```

```
dep(290, raí).
```

```
dep(405, clô).
```

```
dep(405, ary).
```

```
dep(405, noé).
```

Funcionários

Código	Nome	Salário
107	Ana	5.500,00
290	Rui	7.850,00
368	Eva	2.390,00
405	Ivo	4.700,00

Dependentes

Código	Nome
107	Bia
107	Lia
290	Raí
405	Clô
405	Ary
405	Noé

Modelo Relacional e Prolog

Projeção

Seleciona um conjunto de atributos (colunas) de uma tabela.

Exemplo: Quais os nomes e salários dos funcionários?

```
?- func(_,N,S).  
N = ana, S = 5500 ;  
N = rui, S = 7850 ;  
N = eva, S = 2390 ;  
N = ivo, S = 4700  
yes  
  
?- forall(func(_,N,S), writeln([N,S])).  
[ana, 5500]  
[rui, 7850]  
[eva, 2390]  
[ivo, 4700]  
yes
```

Funcionários

Código	Nome	Salário
107	Ana	5.500,00
290	Rui	7.850,00
368	Eva	2.390,00
405	Ivo	4.700,00

Modelo Relacional e Prolog

Seleção

Seleciona um conjunto de tuplas (linhas) de uma tabela, de acordo com uma condição

Exemplo: Quem recebe salário entre 3 e 6 mil reais?

```
?- func(C,N,S), S>=3000, S<=6000.
```

```
C = 107, N = ana, S = 5500 ;
```

```
C = 405, N = ivo, S = 4700
```

```
yes
```

```
?- forall((func(C,N,S), S>=3000, S<=6000),  
          writeln([C,N,S])).
```

```
[107, ana, 5500]
```

```
[405, ivo, 4700]
```

```
yes
```

Funcionários

Código	Nome	Salário
107	Ana	5.500,00
290	Rui	7.850,00
368	Eva	2.390,00
405	Ivo	4.700,00

Modelo Relacional e Prolog

Um relacionamento entre tabelas

É estabelecido com o uso variáveis compartilhadas.

Funcionários

Código	Nome	Salário
107	Ana	5.500,00
290	Rui	7.850,00
368	Eva	2.390,00
405	Ivo	4.700,00

Exemplo: Quem são os dependentes de Ivo?

```
?- forall((func(C,ivo,_), dep(C,N)), writeln(N)).
```

clô

ary

noé

yes

Dependentes

Código	Nome
107	Bia
107	Lia
290	Raí
405	Clô
405	Ary
405	Noé

Modelo Relacional e Prolog

Pouco se sabe da história passada da família **Pinheiro**. Existem alguns registos antigos que indicam que o casal José e Maria criou dois filhos, o João e a Ana. Que a Ana teve duas filhas, a Helena e a Joana, também parece ser verdade, segundo os mesmos registos. Além disso, o Mário é filho do João, pois muito se orgulha ele disso. Estranho também, foi constatar que o Carlos nasceu da relação entre a Helena, muito formosa, e o Mário.

Árvore Genealógica – Família Pinheiro

- a) Utilizando o predicado **progenitor(X,Y)** (ou seja, X é progenitor de Y), represente em Prolog todos os progenitores da família Pinheiro.
- b) Represente em Prolog as relações: **sexo** (masculino ou feminino), **irmã**, **irmão**, **descendente**, **mãe**, **pai**, **avô**, **tio**, **primo**².
- c) Formule em Prolog as seguintes questões:
 - 1. O João é filho do José?
 - 2. Quem são os filhos da Maria?
 - 3. Quem são os primos do Mário?
 - 4. Quantos sobrinhos/sobrinhas com um Tio existem na família Pinheiro?
 - 5. Quem são os ascendentes do Carlos?
 - 6. A Helena tem irmãos? E irmãs?

Árvore Genealógica – Família Pinheiro

pinheiro.pl

% factos

```
progenitor(maria,joao).  
progenitor(jose,joao).  
progenitor(maria,ana).  
progenitor(jose,ana).  
progenitor(joao,mario).  
progenitor(ana,helena).  
progenitor(ana,joana).  
progenitor(helena,carlos).  
progenitor(mario,carlos).
```

```
sexo(ana,feminino).  
sexo(maria,feminino).  
sexo(joana,feminino).  
sexo(helena,feminino).
```

```
sexo(mario,masculino).  
sexo(joao,masculino).  
sexo(jose,masculino).  
sexo(carlos,masculino).
```

Árvore Genealógica – Família Pinheiro

```
irma(X,Y):- progenitor(A,X),  
             progenitor(A,Y),  
             X\==Y,  
             sexo(X,feminino).
```

```
irmao(X,Y):- progenitor(A,X),  
             progenitor(A,Y),  
             X\==Y,  
             sexo(X,masculino).
```

```
descendente(X,Y):- progenitor(X,Y).  
descendente(X,Y):- progenitor(X,A),  
                   descendente(A,Y).
```

```
avo(X,Y):- progenitor(X,A),  
           progenitor(A,Y),  
           sexo(X,masculino).
```

```
mae(X,Y):- progenitor(X,Y),  
           sexo(X,feminino).
```

Árvore Genealógica – Família Pinheiro

```
pai(X,Y):- progenitor(X,Y),
           sexo(X,masculino).

tio(X,Y):- irmao(X,A),
           progenitor(A,Y).

primo(X,Y):-irmao(A,B),
            progenitor(A,X),
            progenitor(B,Y),
            X\==Y.

primo(X,Y):-irma(A,B),
            progenitor(A,X),
            progenitor(B,Y),
            X\==Y.

% questoes:
q1:- progenitor(jose,joao).
```

Árvore Genealógica – Família Pinheiro


```
q1b:- pai(jose,joao).
```

```
q2(X):- mae(maria,X).
```

```
q2b(L):- findall(X,mae(maria,X),L).
```

```
q3(X):- primo(mario,X).
```

```
q3b(L):- findall(X,primo(mario,X),L).
```

```
q3c(L):- findall(X,primo(mario,X),LR),list_to_set(LR,L).
```

```
q4(X):- tio(_,X).
```

```
q4b(L):- findall(X,tio(_,X),LR),list_to_set(LR,L).
```

```
q5(X):- descendente(X,carlos).
```

```
q5b(L):- findall(X,descendente(X,carlos),L).
```

```
q6a(X):- irmao(helena,X).
```

```
q6b(X):- irma(helena,X).
```

Árvore Genealógica – Família Pinheiro