

LISP | Prolog
LISP # #

Programação Funcional e Lógica

1 pt → exercícios mod I e II
1 pt → trabalho* I e II
1 pt → prolog.
28/05

↳ Prolog
↳ exercícios
+ 1 pt

Programação Funcional

- Baseia-se na idéia de calcular
- Paradigma de programação onde
 - Programas consistem em definições de dados e funções
 - Execução de um programa = Avaliação de expressões

Programação Funcional

- Programação com um alto nível de abstração, possibilitando:

calculaTemp(p^1, p^2, p^3, \dots)

- alta produtividade ✓
- programas mais concisos ✓
- menos erros ✓
- provas de propriedades sobre programas ✓

Programação Funcional

- Todos os subprogramas são vistos como funções
 - Eles recebem argumentos e retornam soluções simples.
 - A solução retornada depende apenas da entrada
 - O tempo em que uma função é chamada é irrelevante

Programação Funcional

- Problemas

- “O mundo não é funcional!”

- Esforço inicial não-desprezível

- Implementações ineficientes

- Relevância depende do domínio da aplicação

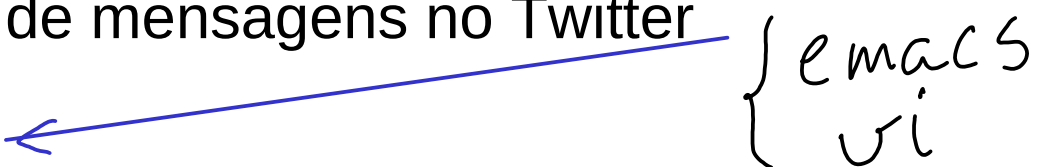
- Mecanismos primitivos de E/S e formatação

- Interface com o usuário

Usos Práticos

- Haskell
 - Ferramentas para manipulação de programas PHP
 - Simulação para a estimativa de riscos em operações financeiras no ABN/AMRO
- Erlang
 - Programação de switches de redes na Ericsson
 - Parte do serviço de chat do Facebook

Usos Práticos

- Scala (linguagem híbrida, parte funcional)
 - Serviço de filas de mensagens no Twitter
- Scheme e LISP 
 - Ensino de programação em várias universidades
- ML
 - Verificação de HW e SW na Microsoft e na Intel




Programação Lógica

- Cálculo de Predicados (Frege, 1879);
- Processamento Simbólico; ←
- Soluções Heurísticas;
- Incluem todas as soluções possíveis.

torce (marcos, palmeiras).

← fato

Usos Práticos

- Sistemas Baseados em Conhecimento (SBCs); 
- Sistemas de Bases de Dados (BDs); 
- Sistemas Especialistas (SEs); 
- Processamento da Linguagem Natural (PLN);
- Modelagem de Arquiteturas Não-Convencionais.

S-VTD-10D

PROLOG

- Conceitos essenciais:
 - **Fatos**: declaram coisas que são incondicionalmente verdadeiras;
 - **Regras**: declaram coisas que podem ser ou não verdadeiras, dependendo da satisfação das condições dadas;
 - **Consultas**: podemos interrogar o programa acerca de que coisas são verdadeiras;

Exemplo - Fatos

progenitor(maria, jose).
progenitor(joao, jose).
progenitor(joao, ana).
progenitor(jose, julia).
progenitor(jose, iris).
progenitor(iris, jorge).

?-progenitor(jose, iris).

Sim

?-progenitor(joao, iris).

Não

?-progenitor(X, iris).

X = jose

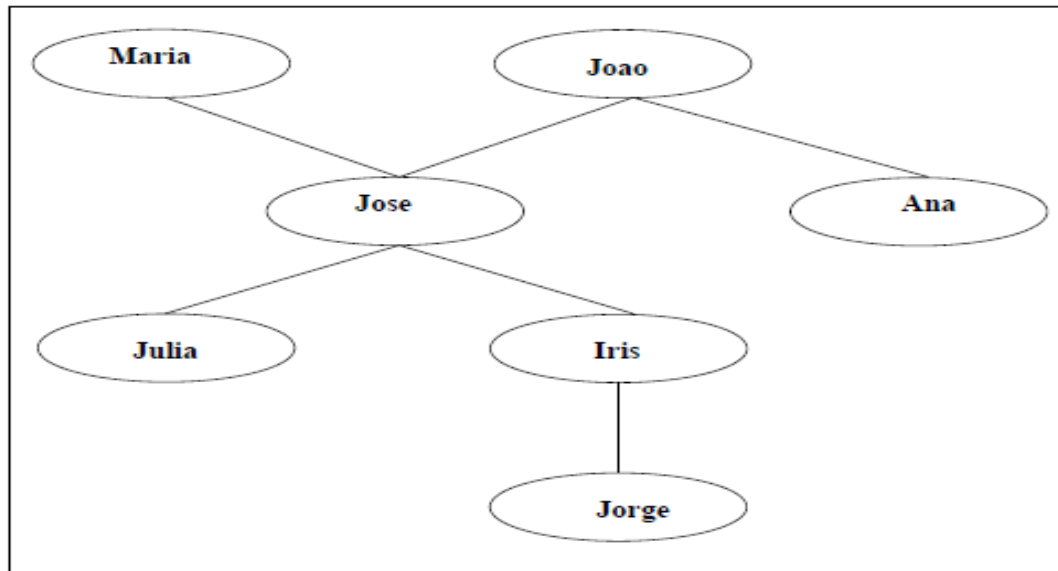


Figura 1: Árvore Genealógica

Exemplo - Regras

- As regras possuem:
 - uma parte de conclusão (o lado esquerdo da cláusula) e,
 - uma parte de condição (o lado direito da cláusula)

filho(Y, X):- progenitor(X, Y).

irmao(X,Y):-progenitor(Z,X), progenitor(Z,Y).

?-irmao(julia, iris).

Sim

?-irmao(X,Ana).

X = Jose

cabeça

corpo

AND

