

# Inteligência Artificial: introdução PROLOG

---

Professores

Alexandre Zamberlan

Ricardo Ferreira de Oliveira

Rodrigo R. V. Goulart

# Objetivos desta aula

- Apresentar os conceitos da programação lógica
  - Prolog;
- Discutir brevemente o Cálculo de Predicado, a base para este tipo de linguagens;
- Apresentar os conceitos básicos do Prolog;
- Tratar alguns problemas do Prolog como linguagem lógica;
- Descrever algumas áreas de aplicação

---

# Sumário

- Introdução
- Cálculo de Predicados
- Visão Geral Programação Lógica
- Origens do Prolog
- Elementos Básicos da linguagem
- Considerações
- Resumo: pontos a lembrar
- Exercícios
- Bibliografia

# Introdução

- Paradigma Lógico é **declarativo** em vez de baseados em procedimentos;
- Expressa programas na forma de **lógica simbólica** (programação lógica);
- Usa um **processo de inferência** lógica para produzir **resultados**;
- **Prolog** única linguagem lógica de uso generalizado
- A **sintaxe** e **semântica** de linguagens lógicas são diferentes das linguagens imperativas.

# Cálculo de Predicados (visão geral)

- Proposição é uma declaração lógica (V ou F)
  - Constituída de objetos (**simples, constantes e variáveis**) e as relações destes
- **Lógica Simbólica** é utilizada na lógica formal:
  - Expressar proposições;
  - Expressar as relações entre elas;
  - Descrever como novas proposições podem ser **inferidas** de outras que se presumem verdadeiras
- Enfim, **Cálculo de Predicados** é a forma de lógica simbólica para programação lógica.

# Cálculo de Predicados (visão geral)

- Proposições são declaradas como fatos ou consultas

homem(jake)

gosta(bob,bife)

- O cálculo de predicados oferece um método para expressar coleções de proposições
- Cláusulas de *Horn* são proposições utilizadas para a resolução (regra de inferência para a demonstração automática de teoremas)

# Cálculo de Predicados (visão geral)

- Cláusulas de *Horn* podem estar somente em duas formas:
  - Proposição atômica no lado esquerdo (**declaram relações**) – com cabeça

$\text{gosta}(\text{bob}, \text{truta}) \rightarrow \text{gosta}(\text{bob}, \text{peixe}) \wedge \text{peixe}(\text{truta})$

- Lado esquerdo vazio (**declaram fatos**) – sem cabeça

$\text{pai}(\text{bob}, \text{jake})$

# Visão Geral Programação Lógica

- Linguagens são declarativas pois **não** possuem **atribuições e instruções de fluxo de controle**
  - São declarações ou proposições (forma de instrução)
- Semântica Declarativa (essência das LPL)
  - O significado de uma proposição é determinado a partir da própria instrução

filho(maria,carlos)

- Os programas nesta linguagem:
  - descrevem a forma como um resultado deve ser computado e não como computá-lo



# Origens do Prolog

- Desenvolvido por:
  - Alain Colmerauer e Phillippe Roussel (University of Aix-Marseille) e
  - Robert Kowalski (University of Edinburgh)
- Em 1981 o governo japonês lança projeto *Fifth Generation Computing Systems* (FGCS) que alvoroça a pesquisadores da europa e dos Estados Unidos.

# Elementos básicos da linguagem

- **Termo** pode ser:
  - Constante – número inteiro ou átomo (cadeia de letras, dígitos e que se inicia com uma letra minúscula)
  - Variável – cadeia de letras, dígitos e que se inicia com uma letra maiúscula
    - Não são vinculadas a tipos
    - São instanciação (vinculação temporária de um valor)
    - Parentes distantes das variáveis das ling. imperativas
  - Estrutura – representam as proposições atômicas
    - *functor(lista de parâmetros)*

# Elementos básicos da linguagem

- **Fatos** (banco de dados): proposições verdadeiras

mulher(beta).

homem(pedro).

mulher(ivone).

homem(jake).

pai(pedro, jake).

pai(pedro, beta).

mae(ivone, jake).

mae(ivone, beta).

# Elementos básicos da linguagem

- **Regras** (hipóteses): o lado direito é a CONDIÇÃO e o esquerdo é a CONSEQUÊNCIA
  - A vírgula indica **AND** (E)  
  
filhoMulher(beta) :- mulher(beta), filho(beta).
  - O **OR** (ou) é indicado pela repetição do predicado  
pais(X, Y) :- mae(X, Y).  
pais(X, Y) :- pai(X, Y).
- A forma geral é:  
consequência :- expressão\_condicional.
  - antepassado(ivone, beta) :- mae(ivone, beta).

# Elementos básicos da linguagem

## ■ Regras: com uso de variáveis

`pais(X, Y) :- mae(X, Y).`

`pais(X, Y) :- pai(X, Y).`

`avo(X, Z) :- pais(X, Y) , pais(Y, Z).`

`irmaos(X, Y) :- mae(M, X) , mae(M, Y) ,  
pai(F, X) , pai(F, Y).`

# Algumas considerações

- Algumas das razões pelas quais adeptos acreditam que Prolog é melhor do que as linguagens imperativas:
  - ❑ O uso de lógica faz com que ocorra menos erros e menos manutenção em seus programas;
  - ❑ O processamento Prolog é paralelo, assim podem tirar (interpretadores) proveito de máquinas com múltiplos processadores;
  - ❑ Boa ferramenta de prototipação, pois possui concisão nos programas que tbém são diminutos

# Convencionais X em Lógica

## **Programas Convencionais**

- Processamento Numérico
- Soluções Algorítmicas
- Estruturas de Controle e Conhecimento Integradas
- Difícil Modificação
- Somente Respostas Totalmente Corretas
- Somente a Melhor Solução Possível

## **Programas em Lógica**

- Processamento Simbólico
- Soluções Heurísticas
- Estruturas de Controle e Conhecimento Separadas
- Fácil Modificação
- Incluem Respostas Parcialmente Corretas
- Incluem Todas as Soluções Possíveis

---

# Outras vantagens

**Além disso, os sistemas de programação em lógica em geral e a linguagem Prolog em particular possuem as seguintes propriedades:**

- Funcionam simultaneamente como linguagem de programação e de especificação,
  - Possuem capacidade dedutiva,
  - Operam de forma não-determinística,
  - Permitem a representação de relações reversíveis,
  - Permitem interpretação declarativa, procedimental e operacional, e
  - São naturalmente recursivos;
-



---

# Principais aplicações

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

# Fatos, regras e consultas

- A programação em Prolog consiste em **estabelecer relações entre objetos e em formular consultas** sobre tais relações.
- Um programa Prolog é formado por cláusulas. Há **três** tipos de cláusulas: **fatos ou assertivas**, **regras ou procedimentos** e **consultas**;
- Uma **relação pode ser especificada por meio de fatos**, que estabelecem as tuplas de objetos que satisfazem a relação, **por meio de regras**, que estabelecem condições para a satisfação das relações, ou **por meio de combinações de fatos e regras** descrevendo a relação;
- Denomina-se **predicado** ao conjunto de fatos e regras empregados para descrever uma determinada relação; Ou seja um predicado é equivalente a uma **RELAÇÃO**.

# Exercícios

- Escreva uma descrição em Prolog de sua árvore genealógica (baseada somente em fatos) retrocedendo até seus avós e incluindo todos os descendentes. Certifique-se de incluir todas as relações.
- Escreva um conjunto de regras para relações familiares, incluindo todas as relações de avós ao longo de duas gerações. Agora, adicione-as aos fatos do problema anterior e elimine tantos fatos quantos puder.

# Exercícios

Considere as seguintes regras na linguagem Prolog:

```
homem( joao ).      mulher( maria ).      amigo( joao, jose ). chefe( joao, jose ).
homem( jose ).      mulher( marta ).      amigo( joao, pedro ).      chefe( marta, tiago ).
homem( pedro ).      mulher( ana ).      amigo( jose, pedro ).      chefe( ana, pedro ).
homem( tiago ).      mulher( clara ).      amigo( pedro, tiago ).      chefe( maria, clara ).
homem( jonas ).      mulher( katia ).      amigo( tiago, jonas ).      chefe( joao, jonas ).
homem( paulo ).      mulher( vera ).      amigo( katia, vera ).      chefe( joao, vera ).
                                chefe( marta, katia ).
                                chefe( ana, paulo ).
chefia( contabilidade, joao ).      primeiro_andar( cadastro ).
chefia( cadastro, marta ).      primeiro_andar( atendimento ).
chefia( producao, ana ).      segundo_andar( contabilidade ).
chefia( atendimento, maria ).      segundo_andar( producao ).

trabalha_em( X, Y ) :- chefia( Y, X ).
trabalha_em( X, Y ) :- chefe( Z, X ), chefia( Y, Z ).
amigo_comum( X, Y ) :- amigo( X, Z ), amigo( Y, Z ).
```

# Exercícios

Qual será a resposta para as seguintes consultas:

- a)trabalha\_em( X, contabilidade ). b)amigo( pedro, X ). c)segundo\_andar( Y ). d)chefia( X, ana ).  
e)amigo\_comum( X, Y ).

Crie uma regra para obter as pessoas que trabalham no primeiro andar. (Sugestão: use a regra trabalha\_em ).

Crie uma regra para obter as pessoas que exercem chefia e são do sexo feminino.

Crie uma regra “amigas” onde ambas devem ser do sexo feminino.

Considere as seguintes regras na linguagem Prolog:

pessoa( maria ). instrumento( violão ). toca( maria, piano ). cordas( violão ).  
pessoa( joao ). instrumento( banjo ). toca( joao, violão ). cordas( banjo ).  
pessoa( armando ). instrumento( piano ). toca( joao, banjo ). cordas( piano ).  
pessoa( rolando ). instrumento( flauta ). toca( armando, banjo ). madeira( flauta ).  
toca( maria, flauta ).

Qual será a resposta para as seguintes consultas:

- a)toca( joao, X ).  
b)toca( JOAO, banjo ).  
c)Crie uma regra para obter as pessoas que tocam instrumento de corda.

# Exercícios

4) Escreva um programa Prolog para representar o seguinte:

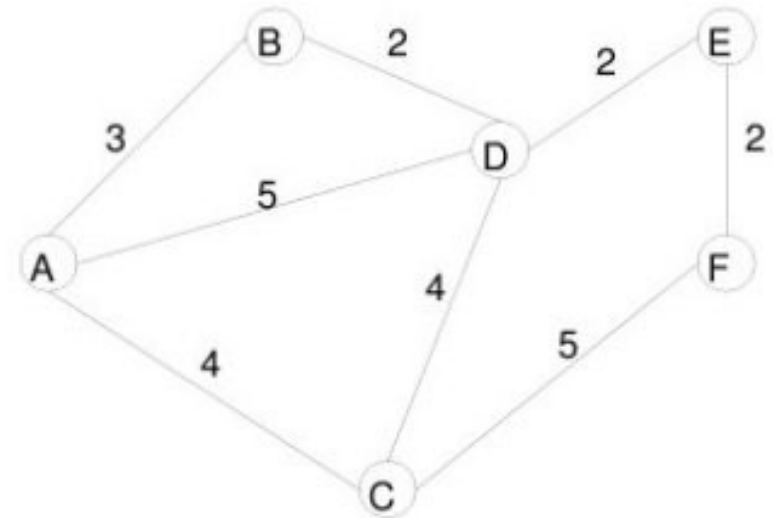
- João nasceu em Pelotas e Jean nasceu em Paris.
- Pelotas fica no Rio Grande do Sul.
- Paris fica na França.
- Só é gaúcho quem nasceu no Rio Grande do Sul.

5) Escreva um programa Prolog para representar o seguinte:

- Os corpos celeste dignos de nota são as estrelas, os planetas e os cometas.
- Vênus é um corpo celeste, mas não é uma estrela.
- Os cometas possuem cauda quando estão perto do sol.
- Vênus está perto do sol, mas não possui cauda.

# Exercícios

- 6) Assuma que os arcos em um grafo expressem custos, como no exemplo abaixo:
- e sejam descritos através de assertivas da forma **arco(R, S, T)**, significando que há um arco de custo T entre os nodos R e S. Por exemplo, `arco(A, B, 3)` descreve um arco de custo 3 entre os nodos A e B.
  - Assuma também que o relacionamento `mais(X, Y, Z)` vale quando  $X+Y=Z$ .
  - Defina o relacionamento `custo(U, V, L)` de forma a expressar que existe um caminho de custo L entre os nodos U e V.



# Referências bibliográficas

- PALAZZO, Luiz A. M. Introdução à programação Prolog. EDUCAT - Editora da Universidade Católica de Pelotas, 1997.
- SEBESTA, Robert W. Conceitos de linguagens de programação. 5º edição. Porto Alegre: Ed. Bookman, 2003.