

GSI010 - Programação Lógica

Organização da disciplina

- ▶ prof. Marcelo Keese Albertini
- ▶ albertini@facom.ufu.br
- ▶ Sala 1A-230
- ▶ <http://www.facom.ufu.br/~albertini/prolog>
 - ▶ Informações sobre a disciplina
 - ▶ plano de aula
 - ▶ notas
 - ▶ trabalhos
 - ▶ datas
 - ▶ slides
 - ▶ avisos
 - ▶ listas de exercícios
 - ▶ exemplos de aula

Avaliação

- ▶ 3 provas teóricas, somando 85 pontos
 - ▶ 1a. prova: 05/07/2013, valendo 25 pontos
 - ▶ 2a. prova: 16/08/2013, valendo 30 pontos
 - ▶ 3a. prova: 20/09/2013, valendo 30 pontos
- ▶ listas de exercícios preparatórios e trabalhos, somando 15 pontos

Referências Bibliográficas

Na biblioteca

- ▶ Clocksin, W. E and Mellish, C.S. Programming in Prolog, 2nd ed. 1984.
- ▶ Arariboia, G. Inteligência Artificial: um curso prático. 1988.
- ▶ Bratko, I. Prolog Programming for Artificial Intelligence, 2nd ed. 1990.

Na Internet

- ▶ Luiz A. M. Palazzo. Introdução a programação Prolog
 - ▶ <http://puig.pro.br/Logica/palazzo.pdf>
- ▶ Eloi L. Favero. Programação em Prolog: uma abordagem prática
 - ▶ <http://www3.ufpa.br/favero>
- ▶ Patrick Blackburn, Johan Bos and Kristina Striegnitz. Learn Prolog Now!
 - ▶ <http://www.learnprolognow.org>

Introdução

- ▶ Conceitos fundamentais da Programação em Lógica
- ▶ Desenvolvimento de aplicações em Prolog
- ▶ Implementação utilizada: Swi-Prolog
 - ▶ <http://www.swi-prolog.org>

O que é Prolog

- ▶ Introdução aos conceitos básicos da programação em lógica e da linguagem Prolog.
- ▶ Fatos e regras.
- ▶ Resolução, unificação, corte e negação.
- ▶ Recursão, controle da execução e estruturas de dados.
- ▶ Semântica declarativa e procedimental.
- ▶ Desenvolvimento de aplicações em Prolog.

Sintaxe do Prolog

Paradigma lógico

- ▶ paradigma declarativo
 - ▶ descrevem o problema, em vez da solução
 - ▶ descreve **o que** deve ser feito, em vez de **como**

Principal linguagem

- ▶ Prolog: Programming in Logic

Sintaxe do Prolog

Porque declarativo?

- ▶ declaração de **fatos**
- ▶ declaração de **regras**
- ▶ execução de código por meio de **consultas**
 - ▶ o prolog avalia consultas de acordo com as regras e fatos previamente definidos
 - ▶ o que não está declarado é falso ou tem resposta negativa
- ▶ conceito de **mundo fechado**

Declarações

Conhecimento

João que amava Teresa que amava Raimundo que amava Maria que amava Joaquim que amava Lili que não amava ninguém. João foi para os Estados Unidos, Teresa para o convento, Raimundo morreu de desastre, Maria ficou para tia, Joaquim suicidou-se e Lili casou com J. Pinto Fernandes que não tinha entrado na história.

Declarações

Conhecimento

João que amava Teresa que amava Raimundo que amava Maria que amava Joaquim que amava Lili que não amava ninguém. João foi para os Estados Unidos, Teresa para o convento, Raimundo morreu de desastre, Maria ficou para tia, Joaquim suicidou-se e Lili casou com J. Pinto Fernandes que não tinha entrado na história.

Transformando em fatos

```
1      amava(joão , teresa ).
2      amava(teresa , raimundo ).
3      amava(raimundo , joaquim ).
4      amava(joaquim , lili ).
5      foi_para(joão , estados_unidos ).
6      foi_para(teresa , convento ).
7      foi_para(raimundo , céu ).
8      foi_para(joaquim , céu ).
9      foi_para(lili , fernandes ).
```

Declarações

Conhecimento

João que amava Teresa que amava Raimundo que amava Maria que amava Joaquim que amava Lili que não amava ninguém. João foi para os Estados Unidos, Teresa para o convento, Raimundo morreu de desastre, Maria ficou para tia, Joaquim suicidou-se e Lili casou com J. Pinto Fernandes que não tinha entrado na história.

Transformando em fatos

```
1      amava(joão , teresa) .  
2      amava(teresa , raimundo) .  
3      amava(raimundo , joaquim) .  
4      amava(joaquim , lili) .  
5      foi_para(joão , estados_unidos) .  
6      foi_para(teresa , convento) .  
7      foi_para(raimundo , céu) .  
8      foi_para(joaquim , céu) .  
9      foi_para(lili , fernandes) .
```

E as letras maiúsculas?

Aplicações da programação lógica

- ▶ Sistemas Baseados em Conhecimento
 - ▶ sistemas que aplicam mecanismos automatizados de raciocínio para a representação e inferência de conhecimento
- ▶ Bancos de Dados “Inteligentes”
 - ▶ sistemas que empregam “agentes” de busca de dados com base em critérios
- ▶ Sistemas Especialistas
 - ▶ sistemas que emulam a especialização humana em algum domínio específico
- ▶ Processamento da Linguagem Natural
 - ▶ usada para desenvolvimento de ferramentas para a comunicação homem-máquina em geral e para a construção de interfaces

Aplicações

- ▶ NASA
 - ▶ Clarissa: interface de voz para execução de procedimentos da Estação Espacial Internacional
- ▶ Indústria de aviação, em soluções de planejamento e escalonamento
 - ▶ coordena 30% do tráfego aéreo do mundo (2010)
 - ▶ <http://www.travelport.com/Careers/Internship>
 - ▶ Estágio 2013: *Experience in one or more of the following languages: .NET (C#, ASP), C++, Prolog, C, Java, HTML/CSS*
- ▶ Prolog “comercial”
 - ▶ Artigo *SICStus Prolog—the first 25 years*, 2010
- ▶ Mineração de dados
 - ▶ busca automática em bancos de dados por padrões e relacionamentos significantes
 - ▶ RDS (Compumine usando SICStus Prolog)

Mudança de paradigma

Desenvolvimento

- ▶ pensar declarativamente, não mais em “procedimentos”
- ▶ modo de abordar problemas é diferente
- ▶ linguagem de alto nível
- ▶ permite a rápida construção de protótipos
- ▶ permite abordar problemas complexos com pouco código
 - ▶ mecanismo de busca de soluções para problema é inerente ao Prolog

Base de conhecimento 1

```
1 mulher(maria).  
2 mulher(joana).  
3 mulher(iolanda).  
4 tocaViolao(joana).  
5 festa.
```

Base de conhecimento 1

```
1 mulher(maria).  
2 mulher(joana).  
3 mulher(iolanda).  
4 tocaViolao(joana).  
5 festa.
```

```
1 ?—
```


Base de conhecimento 1

```
1 mulher(maria).  
2 mulher(joana).  
3 mulher(iolanda).  
4 tocaViolao(joana).  
5 festa.
```

```
1 ?- mulher(maria).  
2
```

Base de conhecimento 1

```
1 mulher(maria).  
2 mulher(joana).  
3 mulher(iolanda).  
4 tocaViolao(joana).  
5 festa.
```

```
1 ?- mulher(maria).  
2 true  
3 ?-
```

Base de conhecimento 1

```
1 mulher(maria).  
2 mulher(joana).  
3 mulher(iolanda).  
4 tocaViolao(joana).  
5 festa.
```

```
1 ?- mulher(maria).  
2 true  
3 ?- tocaViolao(joana).  
4
```

Base de conhecimento 1

```
1 mulher(maria).  
2 mulher(joana).  
3 mulher(iolanda).  
4 tocaViolao(joana).  
5 festa.
```

```
1 ?- mulher(maria).  
2 true  
3 ?- tocaViolao(joana).  
4 true  
5 ?-
```

Base de conhecimento 1

```
1 mulher(maria).  
2 mulher(joana).  
3 mulher(iolanda).  
4 tocaViolao(joana).  
5 festa.
```

```
1 ?- mulher(maria).  
2 true  
3 ?- tocaViolao(joana).  
4 true  
5 ?- tocaViolao(
```

Base de conhecimento 1

```
1 mulher(maria).  
2 mulher(joana).  
3 mulher(iolanda).  
4 tocaViolao(joana).  
5 festa.
```

```
1 ?- mulher(maria).  
2 true  
3 ?- tocaViolao(joana).  
4 true  
5 ?- tocaViolao(maria).  
6
```

Base de conhecimento 1

```
1 mulher(maria).  
2 mulher(joana).  
3 mulher(iolanda).  
4 tocaViolao(joana).  
5 festa.
```

```
1 ?- mulher(maria).  
2 true  
3 ?- tocaViolao(joana).  
4 true  
5 ?- tocaViolao(maria).  
6 false
```

Base de conhecimento 1

```
1 mulher(maria).  
2 mulher(joana).  
3 mulher(iolanda).  
4 tocaViolao(joana).  
5 festa.
```


Base de conhecimento 1

```
1 mulher(maria).  
2 mulher(joana).  
3 mulher(iolanda).  
4 tocaViolao(joana).  
5 festa.
```

```
1 ?—
```

Base de conhecimento 1

```
1 mulher(maria).  
2 mulher(joana).  
3 mulher(iolanda).  
4 tocaViolao(joana).  
5 festa.
```

```
1 ?- tatuada(joana).  
2 ERROR: predicate tatuada/1 not defined.  
3
```

Base de conhecimento 1

```
1 mulher(maria).  
2 mulher(joana).  
3 mulher(iolanda).  
4 tocaViolao(joana).  
5 festa.
```

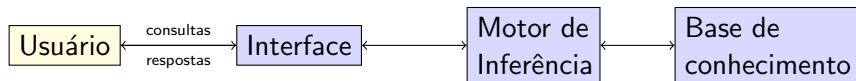
```
1 ?- tatuada(joana).  
2 ERROR: predicate tatuada/1 not defined.  
3  
4 ?- festa.  
5
```

Base de conhecimento 1

```
1 mulher(maria).  
2 mulher(joana).  
3 mulher(iolanda).  
4 tocaViolao(joana).  
5 festa.
```

```
1 ?- tatuada(joana).  
2 ERROR: predicate tatuada/1 not defined.  
3  
4 ?- festa.  
5 true
```

Introdução



O prolog é baseado em:

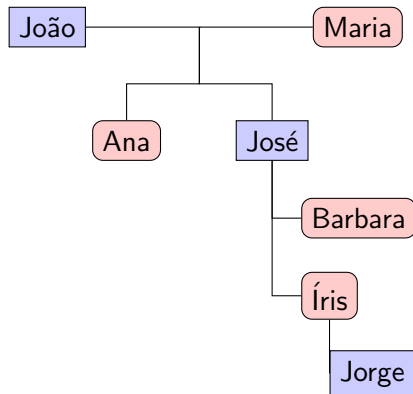
- ▶ Fatos
- ▶ Regras
- ▶ Consultas

Programação em Prolog

Fatos

- ▶ são entendidos como relações entre objetos
- ▶ constituem-se de afirmações que são feitas ao prolog
- ▶ são verdades nas quais o prolog irá basear-se para responder as consultas solicitadas
- ▶ árvore genealógica

Programação em Prolog



Fatos

```
1 progenitor(maria, jose).
```

Programação em prolog

Regras

- ▶ especificação de algo que pode ser verdadeiro se algumas condições forem satisfeitas
- ▶ é composta de duas partes: conclusão (esquerda) e condição (direita)
- ▶ exemplo: relação Filho.

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


Programação em prolog

Consultas

- ▶ questionamentos que serão respondidos avaliando-se os fatos e regras
 - ▶ João é filho de José? $\rightarrow ? - \text{filho}(\text{joão}, \text{josé}).$
 - ▶ Quem é o filho de José? $\rightarrow ? - \text{filho}(X, \text{josé}).$
 - ▶ Quem são os filhos de João? $\rightarrow ? - \text{filho}(X, \text{joão}).$
- ▶ X é uma variável que representa um objeto desconhecido
- ▶ variáveis são escritas com a primeira letra em maiúsculo

Base de conhecimento 2

```
1 progenitor(maria , josé).  
2 progenitor(joão , josé).  
3 progenitor(joão , ana).  
4 progenitor(josé , júlia).  
5 progenitor(josé , íris).  
6 progenitor(íris , jorge).  
7
```

Base de conhecimento 2

```
1 progenitor(maria , josé).  
2 progenitor(joão , josé).  
3 progenitor(joão , ana).  
4 progenitor(josé , júlia).  
5 progenitor(josé , íris).  
6 progenitor(íris , jorge).  
7  
8 filho(X,Y) :- progenitor(Y,X).
```

Dando nomes aos bois:

- fato: `progenitor(maria, josé).`

Base de conhecimento 2

```
1 progenitor(maria , josé).  
2 progenitor(joão , josé).  
3 progenitor(joão , ana).  
4 progenitor(josé , júlia).  
5 progenitor(josé , íris).  
6 progenitor(íris , jorge).  
7  
8 filho(X,Y) :- progenitor(Y,X).
```

Dando nomes aos bois:

- ▶ fato: `progenitor(maria, josé).`
- ▶ regra: `filho(X, Y) :- progenitor(Y, X).`

Base de conhecimento 2

```
1 progenitor(maria , josé).  
2 progenitor(joão , josé).  
3 progenitor(joão , ana).  
4 progenitor(josé , júlia).  
5 progenitor(josé , íris).  
6 progenitor(íris , jorge).  
7  
8 filho(X,Y) :- progenitor(Y,X).
```

Dando nomes aos bois:

- ▶ fato: `progenitor(maria, josé).`
- ▶ regra: `filho(X, Y) :- progenitor(Y, X).`
 - ▶ cabeça (da regra): `filho(X, Y)`

Base de conhecimento 2

```
1 progenitor(maria , josé).  
2 progenitor(joão , josé).  
3 progenitor(joão , ana).  
4 progenitor(josé , júlia).  
5 progenitor(josé , íris).  
6 progenitor(íris , jorge).  
7  
8 filho(X,Y) :- progenitor(Y,X).
```

Dando nomes aos bois:

- ▶ fato: **progenitor(maria, josé).**
- ▶ regra: **filho(X, Y) :- progenitor(Y, X).**
 - ▶ cabeça (da regra): **filho(X, Y)**
 - ▶ corpo (da regra): **progenitor(Y, X)**

Base de conhecimento 2

```
1 progenitor(maria , josé).  
2 progenitor(joão , josé).  
3 progenitor(joão , ana).  
4 progenitor(josé , júlia).  
5 progenitor(josé , íris).  
6 progenitor(íris , jorge).  
7  
8 filho(X,Y) :- progenitor(Y,X).
```

Dando nomes aos boís:

- ▶ fato: **progenitor(maria, josé).**
- ▶ regra: **filho(X, Y) :- progenitor(Y, X).**
 - ▶ cabeça (da regra): **filho(X, Y)**
 - ▶ corpo (da regra): **progenitor(Y, X)**
- ▶ cláusula: um fato ou uma regra (são 7 nessa base)

Base de conhecimento 2

```
1 progenitor(maria , josé).  
2 progenitor(joão , josé).  
3 progenitor(joão , ana).  
4 progenitor(josé , júlia).  
5 progenitor(josé , íris).  
6 progenitor(íris , jorge).  
7  
8 filho(X,Y) :- progenitor(Y,X).
```

Dando nomes aos bois:

- ▶ fato: **progenitor(maria, josé).**
- ▶ regra: **filho(X, Y) :- progenitor(Y, X).**
 - ▶ cabeça (da regra): **filho(X, Y)**
 - ▶ corpo (da regra): **progenitor(Y, X)**
- ▶ cláusula: um fato ou uma regra (são 7 nessa base)
 - ▶ marcado por ponto final

Base de conhecimento 2

```
1 progenitor(maria , josé).  
2 progenitor(joão , josé).  
3 progenitor(joão , ana).  
4 progenitor(josé , júlia).  
5 progenitor(josé , íris).  
6 progenitor(íris , jorge).  
7  
8 filho(X,Y) :- progenitor(Y,X).
```

Dando nomes aos boís:

- ▶ fato: **progenitor(maria, josé).**
- ▶ regra: **filho(X, Y) :- progenitor(Y, X).**
 - ▶ cabeça (da regra): **filho(X, Y)**
 - ▶ corpo (da regra): **progenitor(Y, X)**
- ▶ cláusula: um fato ou uma regra (são 7 nessa base)
 - ▶ marcado por ponto final
- ▶ functor: **filho(X,Y)**

Base de conhecimento 2

```
1 progenitor(maria , josé).  
2 progenitor(joão , josé).  
3 progenitor(joão , ana).  
4 progenitor(josé , júlia).  
5 progenitor(josé , íris).  
6 progenitor(íris , jorge).  
7  
8 filho(X,Y) :- progenitor(Y,X).
```

Dando nomes aos boís:

- ▶ fato: **progenitor(maria, josé).**
- ▶ regra: **filho(X, Y) :- progenitor(Y, X).**
 - ▶ cabeça (da regra): **filho(X, Y)**
 - ▶ corpo (da regra): **progenitor(Y, X)**
- ▶ cláusula: um fato ou uma regra (são 7 nessa base)
 - ▶ marcado por ponto final
- ▶ functor: **filho(X,Y)**
- ▶ argumentos: **filho(X, Y)**

Base de conhecimento 3

```
1 feliz(iolanda).  
2
```

Base de conhecimento 3

```
1 feliz(iolanda).  
2  
3 escuta_musica(maria).  
4 escuta_musica(iolanda):- feliz(iolanda).  
5
```

Base de conhecimento 3

```
1 feliz(iolanda).  
2  
3 escuta_musica(maria).  
4 escuta_musica(iolanda):- feliz(iolanda).  
5  
6 tocaViolao(maria):- escuta_musica(maria).  
7 tocaViolao(iolanda):- escuta_musica(iolanda).
```

São cinco cláusulas na base de conhecimento.

Base de conhecimento 3

```
1 feliz(iolanda).  
2  
3 escuta_musica(maria).  
4 escuta_musica(iolanda):- feliz(iolanda).  
5  
6 tocaViolao(maria):- escuta_musica(maria).  
7 tocaViolao(iolanda):- escuta_musica(iolanda).
```

São cinco cláusulas na base de conhecimento.

São três **predicados**, feliz, escuta_musica e ...

```
1 ?- tocaViolao(maria).  
2
```

Base de conhecimento 3

```
1 feliz(iolanda).  
2  
3 escuta_musica(maria).  
4 escuta_musica(iolanda):- feliz(iolanda).  
5  
6 tocaViolao(maria):- escuta_musica(maria).  
7 tocaViolao(iolanda):- escuta_musica(iolanda).
```

São cinco cláusulas na base de conhecimento.

São três **predicados**, feliz, escuta_musica e ...

```
1 ?- tocaViolao(maria).  
2  
3 true  
4
```

Base de conhecimento 3

```
1 feliz(iolanda).  
2  
3 escuta_musica(maria).  
4 escuta_musica(iolanda):- feliz(iolanda).  
5  
6 tocaViolao(maria):- escuta_musica(maria).  
7 tocaViolao(iolanda):- escuta_musica(iolanda).
```

São cinco cláusulas na base de conhecimento.

São três **predicados**, feliz, escuta_musica e ...

```
1 ?- tocaViolao(maria).  
2  
3 true  
4  
5 ?- tocaViolao(iolanda).  
6
```


Base de conhecimento 3

```
1 feliz(iolanda).  
2  
3 escuta_musica(maria).  
4 escuta_musica(iolanda):- feliz(iolanda).  
5  
6 tocaViolao(maria):- escuta_musica(maria).  
7 tocaViolao(iolanda):- escuta_musica(iolanda).
```

São cinco cláusulas na base de conhecimento.

São três **predicados**, feliz, escuta_musica e ...

```
1 ?- tocaViolao(maria).  
2  
3 true  
4  
5 ?- tocaViolao(iolanda).  
6  
7 true
```

Base de conhecimento 4

```
1 feliz(vicente).
2 escuta_musica(bruno).
3 tocaViolao(vicente):- escuta_musica(vicente),
4                          feliz(vicente).
5 tocaViolao(bruno):- feliz(bruno).
6 tocaViolao(bruno):- escuta_musica(bruno).
7 tocaViolao(bruno):- feliz(bruno); escuta_musica(bruno).
8 tocaViolao(vicente):- \+feliz(maria).
```

Leia-se

- ▶ operador “dois pontos hífen” “:-”: **se**
- ▶ operador “vírgula” “,”: “e” (conjunção)
- ▶ operador “ponto-e-vírgula” “;”: “ou” (disjunção)
- ▶ operador “\+”: negação

Variáveis

```
1 mulher(maria).  
2 mulher(joana).  
3 mulher(iolanda).  
4 ama(vicente, maria).  
5 ama(joão, maria).  
6 ama(abobrinha, coelhinho).  
7 ama(coelhinho, abobrinha).
```

```
1 ?- mulher(X).  
2 X = maria
```

Variáveis

```
1 mulher(maria).  
2 mulher(joana).  
3 mulher(iolanda).  
4 ama(vicente, maria).  
5 ama(joão, maria).  
6 ama(abobrinha, coelhinho).  
7 ama(coelhinho, abobrinha).
```

```
1 ?- mulher(X).  
2 X = maria;  
3 X = joana
```

Variáveis

```
1 mulher(maria).  
2 mulher(joana).  
3 mulher(iolanda).  
4 ama(vicente, maria).  
5 ama(joão, maria).  
6 ama(abobrinha, coelhinho).  
7 ama(coelhinho, abobrinha).
```

```
1 ?- mulher(X).  
2 X = maria;  
3 X = joana;  
4 X = iolanda
```

Variáveis

```
1 mulher(maria).  
2 mulher(joana).  
3 mulher(iolanda).  
4 ama(vicente, maria).  
5 ama(joão, maria).  
6 ama(abobrinha, coelhinho).  
7 ama(coelhinho, abobrinha).
```

```
1 ?- mulher(X).  
2 X = maria;  
3 X = joana;  
4 X = iolanda.  
5 ?
```

Operador ; significa “ou”

Base de conhecimento 5

```
1 mulher(maria).  
2 mulher(joana).  
3 mulher(iolanda).  
4 ama(vicente, maria).  
5 ama(joão, maria).  
6 ama(abobrinha, coelhinho).  
7 ama(coelhinho, abobrinha).
```

Usando o operador vírgula ,

```
1 ?- ama(abobrinha,X), mulher(X).  
2 false
```

Base de conhecimento 6

```
1 mulher(maria).  
2 mulher(joana).  
3 mulher(iolanda).  
4 ama(vicente , maria).  
5 ama(joão , maria).  
6 ama(abobrinha , coelhinho).  
7 ama(coelhinho , abobrinha).  
8  
9 tem_ciumes(X,Y):- ama(X,Z) , ama(Y,Z) .
```

```
1 ?- tem_ciumes(joão ,W).  
2 W = vicente ;  
3 W = joão ;  
4 false .  
5 ?-
```


Base de conhecimento 7

Fatos

```
1 gosta(joao , peixe) .  
2 gosta(joao , maria) .  
3 gosta(maria , livro) .  
4 gosta(pedro , livro) .  
5 gosta(maria , flor) .  
6 gosta(maria , vinho) .
```

Qual é o resultados de:

```
1 ?- gosta(maria ,X) .  
2 ?- gosta(X , livro) .  
3 ?- gosta(Quem , Oque) .  
4 ?- gosta(X,Y) .  
5 ?- gosta(X,X) .  
6 ?- gosta(_a , _b) .  
7 ?- gosta(A , peixe) .
```

Referências

- ▶ Luis, A. M. Palazzo, Introdução à programação prolog, Educat, 1997
- ▶ Slides da Profa. Solange - ICMC - USP