Disciplina:
Matemática Discreta
Docente:
Prof. Dr. Cícero Costa Quarto

Prolog

Programação em Lógica *ou* *Programming in Logic*

Composição de Equipe:
Claudenilson da Silva Morais
Francyellen Chaves Dutra
Luíz Otávio Vieira Silva
Kaik Klydson Costa Alves
Slatter Júnior Cardozo Rodrigues

O que é e como surge o Prolog?

Conceituação e contextualização histórica

- O **Prolog**, ou **Pro**gramming in **Log**ic, é uma linguagem de programação lógica criada e aperfeiçoada no fim da década de 60 e início da década de 70 entre as Universidades de Aix-Marseille, França, e de Edimburgo, Reino Unido, pelos pesquisadores Alain Colmerauer, Philippe Roussel, Robert Kowalski e David Warren ;

- Ao contrário da linguagem LISP (outra linguagem específica para o desenvolvimento de sistemas de IA), mais utilizada nos EUA, teve maior utilização na Europa e Japão;

- É considerada uma linguagem de programação *declarativa*, isto é, "a descrição do problema é baseada em 'o que' realizar, em vez de 'como' realizar", na contramão de paradigmas de linguagem *procedurais* ou *imperativos.*

Como o Prolog *pensa*?

Fatos, regras e *queries* (consultas)

- O Prolog se constitui a partir de **uma coleção de fatos** (base de dados) e **regras** (relações lógicas), os quais resultam ***na descrição de um determinado problema*** (Barbosa; Cunha, 2006, p. 4);
- O fato, necessariamente, deve ser descrito no programa a partir dos seguintes critérios:
  - Início com letra minúscula (nome do predicado);
  - Pode contar com argumentos entre parênteses;
  - Termina com . (ponto final);
  - Argumentos: nomes -> minúsculo;
  - Variáveis: nomes -> maiúsculo.

Exemplificação de fatos em Prolog

- **Exemplos:**

% Fatos simples

gato(mimi).

cachorro(rex).

cor(ceu, azul).

idade(joao, 25).

gostar(maria, sorvete).

- **Lê-se:**

“mimi é um gato”;

“rex é um cachorro”;

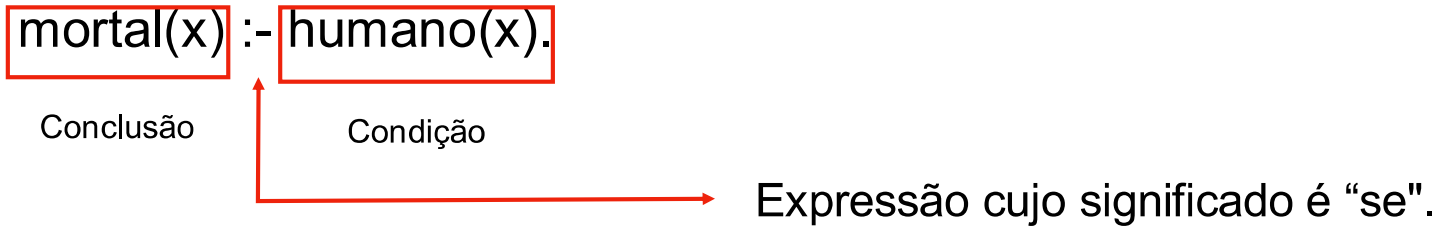
“João tem 25 anos” etc.

Regras em Prolog

- Para Barbosa e Cunha (2006), **as regras Prolog são descrições de predicados por meio de condicionais**. Um exemplo é o predicado *pai(x)*, significando “x é pai” através da regra pai(X) :- prole(X,_), homem(X) (ou *x é pai se x possui ao menos um filho*);

- Pode-se, simplesmente, assumir que **uma regra Prolog diz que “x é verdade SE outras condições forem verdadeiras”;**

- A estrutura de uma regra pode ser exemplificada do seguinte modo:



com significado: “mortal(x) é verdade SE humano(x) for verdade".

Exemplos e mini-glossário de expressões

- % “X pode voar **SE** X é um pássaro”, ou

  voa(x) **:-** passaro(x) (grifo nosso).

- % “X é avô de Z **SE** X é pai de Y

  % “**E** Y é pai de Z”, ou

  avo(X, Z) **:-** pai (X, Y), pai(Y, Z).

- **Se**, portanto, pode ser representado graficamente com **:-** (ponto e hífen), enquanto **E** pode ser representado com **,** (vírgula).

- A estrutura da regra em Prolog, conforme Oliveira e Da Silva (2013), pode ainda ser expressa na separação de estruturas de **cabeça** e **corpo** por **:-** (pelo menos no quadro da Sintaxe de Edimburgo), sendo:

Cabeça a representação do predicado que se pretende provar, juntamente aos seus argumentos e;

Corpo um conjunto de objetivos que devem ser satisfeitos a fim de que o predicado presente na cabeça seja verdadeiro.

Assim:

Cabeça :- corpo.

Queries ou consultas em Prolog

- Para Oliveira e Da Silva (2013), **o sistema Prolog aceita os fatos e regras como um conjunto de axiomas e a consulta do usuário como um teorema a ser provado**.

- A tarefa do sistema no procedimento de consulta é, portanto, **demonstrar que o teorema pode ser provado *com base nos axiomas representados pelo conjunto das cláusulas que constituem o programa***;

- Operacionalmente, o consulente inicia a pergunta com ?-, sendo respondido com os valores *true* ou *false.*

Exemplos

- **Base de dados qualquer (fatos e regras):**

  humano(socrates).

  humano(aristoteles).

  humano(platao).

  mortal(X) :- humano(X).

- **Consultas e respostas:**

  ?- humano(socrates).

  true.

  ?- humano(batman).

  false.

  ?- mortal(platao).

  true.

Variáveis, unificação e *backtracking* em Prolog

- As **variáveis** em Prolog representam valores desconhecidos, começando com letras maiúsculas ou com o caractere "_". Exemplo: X, Nome, Idade, _Pessoa.
- Considere os fatos:

pai(joao, maria).

pai(joao, pedro).

- Consulta:

?- pai(joao, X).

- Resultado:

X = maria;

X = pedro.

Nesse caso, a variável X recebe os valores que tornam a consulta verdadeira.

Unificação

- A **unificação** é o mecanismo utilizado por Prolog para verificar se dois termos podem se tornar iguais. Exemplo:

?- X = maria.

- Resultado:

X = maria.

- Outro exemplo:

?- gosta(ana, Y) = gosta (ana, pizza).

- Resultado:

Y = pizza.

- **A unificação falha quando os termos não podem ser tornados iguais:**

?- gosta(ana, pizza) = gosta(carlos, pizza).

- Resultado:

False.

- **A unificação, portanto, é a base do processo de busca de respostas em Prolog.**

Backtracking

- O ***backtracking*** é o mecanismo que permite ao Prolog procurar automaticamente outras soluções quando uma tentativa falha ou quando o usuário solicita mais respostas.
- Exemplo:

gosta(maria, pizza).

gosta(maria, lasanha).

gosta(maria, hamburguer).

- Consulta:

?- gosta(maria, X).

- Primeira resposta:

X = pizza

- Ao pressionar ;, o Prolog faz backtracking e procura outra possibilidade:

X = lasanha;

X = hamburguer;

False.

Exemplos

- **Base de conhecimento**:

homem(joao).

homem(pedro).

mulher(maria).

pai(joao, maria).

pai(joao, pedro).

- **Consulta**:

?- pai(joao, filho).

- **Processo:** Prolog procura fatos que correspondam a pai(joao, filho).

- **Encontra:**

pai(joao, maria).

- **Unificação:**

filho = maria

- Se o usuário pedir outra resposta (;), ocorre *backtracking.*
- O Prolog retorna ao ponto anterior e tenta outra alternativa:

pai(joao,pedro).

- **Nova unificação:**

filho = pedro

Listas em Prolog

- **Lista** é uma sequência ordenada de elementos, sendo **elementos** de diferentes tipos, como *números, átomos, variáveis* ou outras listas;
- Os elementos de uma lista devem ser separados por vírgulas e delimitados por colchetes. Exemplos:

```
[1, 2, 3, 4] % lista simples

[a, b, c] % lista de átomos

[[1, 2], [3, 4]] % lista contendo outras listas

[H | T] % H = cabeça, T = cauda

?- [H | T] = [10, 20, 30]

H = 10,

T = [20, 30]
```

Principais comandos para Lista

- Verifica se X pertence à lista

member(2,[1,2,3,4,5)].

True

- Retorna o tamanho da lista

length([1,2,3,4,5],N).

N = 5

- Concatenar as listas

append([1,2], [3,4,5],X).

X = [1,2,3,4,5]

- Inverte a lista

reverse([1,2,3,4,5],X).

X = [5,4,3,2,1]

- Ordena e remove duplicados

sort([3,1,2,4,3,1],X).

X = [1,2,3,4]

Utilização e Aplicações

- **IA - Inteligência Artificial**, em específico em sistemas especialistas e raciocínio automático. Exemplo: diagnóstico médico por regras lógicas;
- **NL - Linguagem Natural**, em específico análise de frases, gramáticas e *chatbots*. Prolog, por sua vez, tem suporte nativo a gramáticas (DCG);
- **Banco de Dados**, em específico consultas complexas e bancos dedutivos. Exemplo: Datalog é baseado em Prolog;
- **VF - Verificação Formal**, a partir da tentativa de provar matematicamente que sistemas de *software* ou *hardware* estão corretos.

Muito obrigado(a) pela atenção!

Referências Bibliográficas:

BARBOSA, A. A.; CUNHA, J. F. **Introdução à Programação em Lógica**. Universidade Federal de Campina Grande - UFCG, 2006.

OLIVEIRA, G. S., DA SILVA, A. F.; **Prolog:** a linguagem, a máquina abstrata de Warren e Implementações. Rita: Maringá, Vol. 20, Nº 2, 2013.

Composição de Equipe:

Claudenilson da Silva Morais
Francyellen Chaves Dutra
Luíz Otávio Vieira Silva
Kaik Klydson Costa Alves
Slatter Júnior Cardozo Rodrigues