

## 1 Breve Histórico

Segundo (Bratko, 1990), a idéia de usar lógica como formalismo executável em um computador surgiu nos anos 70. Os primeiros desenvolvedores da linguagem foram Robert Kowalski, responsável pela parte teórica: o projeto, Maarten, que elaborou uma demonstração experimental da idéia, e Alain Colmerauer, que realizou a implementação do primeiro interpretador Prolog (PROgramação LÓGica).

A popularidade da linguagem é atribuída a David Warren que fez uma versão mais eficiente do Prolog.

## 2 Características

“O Prolog é uma linguagem de programação que gira em torno de um conjunto pequeno de mecanismos, tais como: casamento de padrões (*pattern matching*), estruturas de dados baseadas em árvores e retrocesso (*backtracking*) automático.” (Bratko, 1990),

Dentre as suas características, pode-se destacar:

- Orientada a processamento simbólico;
- Implementa lógica como linguagem de programação;
- Permite a definição de programas inversíveis, ou seja, programas que não distinguem entre argumentos de entrada e saída. Permite a definição de um programa com mais de uma finalidade que pode ser chamado de formas diferentes;
- Permite respostas alternativas;
- Suporta estruturas de dados similares a registros ou listas;
- Permite recuperação dedutiva de informação;
- Suporta codificações recursivas e dispensa os mecanismos tradicionais de controle como *for* e *while*;
- Representa programas e dados com o mesmo formalismo.

## 3 Sintaxe

A sintaxe da linguagem é muito simples. Ela possui um alfabeto básico e essencialmente três tipos de objetos: constantes, variáveis e predicados.

### 3.1 Alfabeto

O alfabeto da linguagem é constituído de alfanuméricos e de alguns caracteres especiais. Dentre os caracteres especiais, pode-se destacar:

- Parênteses () – Usados para formar um predicado.
- Ponto (.) – Usado ao final de toda linha.
- Ponto-e-vírgula (;) – tem a função do operador lógico **ou**. Algumas versões implementam a barra vertical com esta função (|).
- Vírgula (,) – tem a função do operador lógico **e**.
- Exclamação (!) – representa o operador **cut**.
- Dois Pontos seguido de hífen (:-) – representa o operador de implicação. Normalmente utilizado em regras Prolog.
- Underline (\_) : variável anônima.

### 3.2 Objetos Prolog

Sintaticamente todos os objetos em Prolog são **termos**. O conjunto de termos Prolog é o menor conjunto que satisfaz às seguintes condições:

- Toda constante é um termo;
- Toda variável é um termo;
- Se  $t_1, t_2, \dots, t_n$  são termos e  $f$  é um nome de função, então  $f(t_1, t_2, \dots, t_n)$  também é um termo.

#### 3.2.1 Variáveis

As variáveis sempre devem começar com letras maiúsculas. Não há declaração de variáveis e nem qualquer informação adicional sobre seus tipos. O identificador de uma variável não deve conter caracteres especiais, exceto o "\_" (underscore). Uma variável identificada apenas pelo "\_" é dita "variável anônima" e é utilizada quando não há interesse por algum objeto de um predicado.

Exemplos de variáveis: Nome, N, Ana, NOME, \_.

#### 3.2.2 Constantes

As constantes são constituídas pelos valores numéricos, strings e caracteres.

Os caracteres e as strings devem obrigatoriamente iniciar por letra minúscula e não podem conter caracteres especiais (exceto o *underline* ou no caso de estarem entre apostrofes). Para defini-las com letras maiúsculas ou com espaços em branco, é necessário colocar a string ou o caracter entre apóstrofes. As strings ainda podem ser "ligadas" pelo caracter especial *underline*.

Os tipos numéricos implementados compreendem os inteiros e ponto-flutuante. Dentre os operadores aritméticos, pode-se destacar: `is` (unifica uma variável a um valor), `+` (adição), `*` (multiplicação), `-` (subtração), `/` (divisão), `//` (divisão inteira) e `mod` (resto da divisão inteira). Dentre os operadores relacionais, citam-se: `>=` (maior ou igual), `<=` (menor ou igual), `/=` (diferente), etc.

Exemplos: `a`, `ana`, `'Ana'`, `12`, `12.45`, `joao_batista`, `'João Batista'`.

### 3.2.3 Predicados

Predicados ou estruturas são objetos que possuem vários componentes. Os próprios componentes, por sua vez, podem ser também estruturas. Por exemplo, uma data pode ser vista com o uma estrutura de 3 componentes: dia, mês e ano. Mesmo que as estruturas sejam formadas por diversos componentes, elas são tratadas no programa como objetos simples.

Para combinar os componentes em uma estrutura é necessário empregar um *functor*. *Functor* é um símbolo funcional (um nome de função) que permite agrupar diversos objetos em um único objeto estruturado.

Cabe ressaltar que o nome do predicado segue as regras de definição de uma constante string, ou seja, deve ser iniciar por letra minúscula, não pode conter caracteres especiais, exceto o *underline*.

Exemplos:       pai(joao, maria).  
                  livro(assunto(prolog), autor(bratko)).  
                  conceito(Nota, aprovado) :- Nota >=6.

## 4 Unificação

A operação mais importante entre 2 termos Prolog é a Unificação. A unificação é responsável pelo casamento de padrões. É através dela, por exemplo, que um valor é associado a uma variável. Entretanto, ela não deve ser confundida com a operação de atribuição usual das linguagens procedurais.

A seguir, apresentam-se as regras de unificação, que determinam que objetos unificam com outros objetos. Unificam:

- *Variáveis com Variáveis* : Uma variável sempre unifica com outra variável, independentemente do seu identificador.
- *Variáveis com Constantes* : Uma variável sempre unifica com uma constante.
- *Variáveis com Estruturas*: Uma variável sempre unifica com uma estrutura.
- *Constantes com Constantes*: Uma constante só unifica com outra se forem idênticas.
- *Estruturas com Estruturas*: Uma estrutura só unifica com outra quando possuem o mesmo nome de função e a mesma quantidade de argumentos. Para a unificação dos argumentos valem todas as regras já definidas.

**É importante ressaltar que uma vez que uma variável X de uma cláusula unifica com um determinado valor, todas as ocorrências de X nesta mesma cláusula unificarão com o mesmo valor.**

*Exemplo* : livro(assunto(prolog), autor(bratko)) **unifica com** livro (X, Y)

*X* será instanciado com *assunto(prolog)*

*Y* será instanciado com *autor(batko)*

### Exercícios:

- 1) Associe a coluna da direita com a da esquerda, identificando os termos Prolog, bem como, as representações incorretas.

<b>( P )</b>	<b>Predicado</b>	( )	Marcelo
<b>( V )</b>	<b>Variável</b>	( )	Endereco
<b>( C )</b>	<b>Constante</b>	( )	tipo
<b>( E )</b>	<b>Erro</b>	( )	120.70
		( )	x
		( )	Movimentação(X,dir):- X < 20.
		( )	linguagem(prolog).
		( )	linguagem(c, paradigma(procedural)).
		( )	'Antônia'
		( )	João
		( )	a
		( )	—
		( )	-
		( )	'-'
		( )	luiz serra

- 2) Os termos prolog abaixo unificam ? Se a unificação ocorrer com sucesso, determine o valor das variáveis. O símbolo ~ foi usado para representar a possibilidade de unificação.

- `data(8, 3, 2001) ~ data(X, 3, 01)`
- `data(8, 3, 2001) ~ dia(X, Y, 2001)`
- `data(D,M,A) ~ data (20, Maio, 2003)`
- `data(D,M,A) ~ data (10, junho, 1979)`
- `data(D, junho, A) ~ data( 2, Mes, decada(70))`
- `livro(assunto('Inteligência Artificial'), autor('Elaine Rich')) ~ livro (A, A)`
- `livro(assunto('Inteligência Artificial'), autor('Elaine Rich')) ~ livro (As, Au)`
- `livro(assunto('Inteligência Artificial'), autor('E.Rich')) ~ livro (assunto(A), autor(N))`
- `peessoa(Ana, 18, informatica) ~ pessoa(x, 18, Y)`
- `peessoa(ana, 18, informatica) ~ pessoa(x, 18, Y)`
- `peessoa(ana, informatica) ~ pessoa(x, 18, Y)`
- `peessoa(ana, 18, informatica) ~ pessoa(Nome, __, Curso)`
- `peessoa(ana, 18, informatica) ~ pessoas(__, __, Curso)`
- `computador(pentiumIII, 233) ~ computador(pentiumIII,233,ram64)`
- `curso(hardware, torres) ~ curso( __, X).`

- p. objeto(bola, vermelha, grande) ~ objeto(X, azul, Y).
- q. eventos(simposios, congresso, workshops) ~ evento (X, \_, Y).
- r. pessoa (adriano, 21) ~ pessoa(Andre,21).
- s. pessoa(adriano,21,sexo(masc), curso(computacao)) ~ pessoa(adriano, \_, sexo(X), curso(Y)).
- t. livro(linux, editora(makron), autor(richard)) ~ livro(linux, X, Y).
- u. endereco(andradas, 1001, andar(7), fone('221-4589'), contato(maria)) ~ endereco(\_,\_,\_,X,Y)

3) Represente em Prolog as sentenças abaixo:

- a. O Windows é um sistema operacional.
- b. O Linux também é um sistema operacional.
- c. Todo sistema operacional é um software porém básico.
- d. O Word é um editor de textos.
- e. O Excel é uma planilha eletrônica.
- f. Todo o editor de textos ou planilha eletrônica é um software porém aplicativo.
- g. Quais são os softwares básicos ?
- h. Quais os nomes dos softwares ?
- i. Word é um sistema operacional ?
- j. Excel é um aplicativo ?

4) Considere a base Prolog abaixo que representa as relações de família de "Joãozinho".

```
progenitor(antonio,joaozinho).
progenitor(maria,joaozinho).
progenitor(antonio,luis).
progenitor(maria,luis).
progenitor(antonio,cristina).
progenitor(maria,cristina).
progenitor(carlos,antonio).
progenitor(fernanda,antonio).
progenitor(pedro,maria).
progenitor(mariana,maria).
progenitor(jose,mariana).
progenitor(sandra,mariana).
progenitor(carlos,alberto).
progenitor(fernanda,alberto).
progenitor(carlos,marcia).
progenitor(fernanda,marcia).

sexo(joaozinho,masc).
sexo(antonio,masc).
```

sexo(maria,fem).

...

Defina os predicados:

- a. pai
- b. mãe
- c. irmão
- d. irmã
- e. tio
- f. tia
- g. avô
- h. avó
- i. tio-avô
- j. tia-avó
- k. bisavô
- l. bisavó
- m. ancestral