

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO SUPERIOR DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Dionatan Eduardo Correa Rodrigues

PARADIGMAS DE PROGRAMAÇÃO
PROLOG

Santa Maria, RS
2024

SUMÁRIO

1	APRESENTAÇÃO DE CONCEITOS	3
1.1	FATOS	3
1.2	REGRAS.....	3
1.3	PERGUNTAS.....	4
2	SINTAXE	5
	REFERÊNCIAS BIBLIOGRÁFICAS	6

1 APRESENTAÇÃO DE CONCEITOS

Prolog é uma linguagem descritiva e prescritiva. Ao mesmo tempo que descreve-se o que deve ser feito, prescreve-se como isto deve ser feito. Ela lida com objetos e relações entre eles. A palavra “objeto” não tem o mesmo sentido que em orientação a objetos, pois os objetos Prolog não têm métodos e não há herança. Em Prolog, objetos são apenas coisas sobre as quais queremos raciocinar. Um programa em Prolog é composto de fatos sobre certos objetos, regras de inferência e perguntas sobre os objetos.

Dizemos a Prolog certos fatos e regras, e depois fazemos perguntas sobre estes fatos e regras. Por exemplo, podemos informar Prolog sobre irmãos e depois perguntar se Maria e Joana são irmãs. Prolog responderá sim ou não em função do que lhe dissermos. Prolog na verdade faz muito mais do que responder sim ou não: a linguagem permite usar o computador como um estrutura de fatos e regras, e proporciona meios de fazer inferências, indo de um fato a outro, e achando os valores das variáveis que podem levar a conclusões lógicas. Prolog é geralmente usada como uma linguagem interpretada.

1.1 FATOS

Aqui temos alguns exemplos de como se informam fatos a Prolog:

<code>gosta(pedro, maria).</code>	Pedro gosta de Maria
<code>gosta(maria, pedro).</code>	Maria gosta de Pedro
<code>valioso(ouro).</code>	Ouro é valioso
<code>mulher(jane).</code>	Jane é mulher
<code>possui(jane, ouro).</code>	Jane possui ouro
<code>pai(pedro, maria).</code>	Pedro é pai de Maria
<code>entrega(romeu, livro, maria).</code>	Romeu entrega o livro a Maria

Observe que: nomes de relações e objetos iniciam-se com letra minúscula; o nome da relação vem primeiro, depois vem a lista de objetos separados por vírgula e envolta em parênteses; o ponto final é obrigatório ao final do fato.

1.2 REGRAS

Uma regra é uma afirmação geral sobre objetos e seus relacionamentos. Por exemplo, suponha que queremos representar a seguinte dependência entre fatos: Pedro gosta de todo mundo que gosta de vinho, o que pode ser reescrito como: Pedro gosta de X se X gosta de vinho. Em Prolog, regras consistem de uma cabeça e um corpo. A cabeça e o corpo são conectados pelo símbolo “:-” formado por dois pontos e hífen. O “:-” pronunciase “se”. A dependência acima seria escrita como:

```
gosta(pedro, X) :- gosta(X, vinho).
```

Note que regras também terminam com ponto final. A cabeça desta regra é `gosta(pedro, X)`. A cabeça de uma regra descreve o que está sendo definido. O corpo, no caso `gosta(X,`

vinho), é uma conjunção de metas que devem ser satisfeitas para que a cabeça seja considerada verdadeira. Por exemplo, podemos tornar Pedro mais exigente sobre o que ele gosta adicionando mais metas ao corpo da regra:

```
gosta(pedro, X) :- gosta(X, vinho), gosta(X, menta).
```

ou, em outras palavras, “Pedro gosta de qualquer um que goste de vinho e de menta”. Ou então, supondo que Pedro gosta de mulheres que gostam de vinho:

```
gosta(pedro, X) :- mulher(X), gosta(X, vinho).
```

Note que a mesma variável X ocorre três vezes na regra. Dizemos que o escopo de X é a regra toda. Isto significa que quando X for instanciada, as três ocorrências terão o mesmo valor.

1.3 PERGUNTAS

Uma pergunta em Prolog tem a forma:

```
?- possui(maria, livro).
```

Estamos perguntando se Maria possui o livro. Prolog tenta unificar o fato da pergunta com os fatos do banco de dados. Dois fatos se unificam se têm o mesmo predicado e os mesmos argumentos na mesma ordem. Se Prolog achar um fato que unifica com a pergunta, vai responder “sim”. Caso contrário, responderá “não”. Perceba que a resposta “não” em Prolog não significa necessariamente que o fato não é verdadeiro, mas simplesmente que Prolog não consegue provar o fato a partir de seu banco de dados. Confira isto no seguinte exemplo. Considere o banco de dados:

```
humano(socrates).  
humano(aristoteles).  
ateniense(socrates).
```

e a pergunta:

```
?- ateniense(aristoteles).
```

Embora seja verdade que Aristóteles tenha sido ateniense, não se pode provar isto a partir dos fatos dados.

2 SINTAXE

Em Prolog há apenas um tipo, chamado de termo, que engloba todas as construções sintáticas da linguagem. Um termo pode ser uma constante, uma variável, ou uma estrutura. As constantes podem ser átomos ou números, onde um átomo indica um objeto ou uma relação. Nomes de objetos como maria, livro, etc. são átomos e nomes de átomos sempre começam com letra minúscula. Em relação a números, Prolog acompanha as outras linguagens, permitindo inteiros positivos e negativos, números em ponto flutuante usando ponto decimal e opcionalmente expoente de dez. Alguns exemplos de números válidos:

0, 1, -17, 2.35, -0.27653, 10e10, 6.02e-23

Por sua vez, as variáveis têm nomes cujo primeiro caractere é uma letra maiúscula ou o sinal de sublinhado. Estas últimas são chamadas de variáveis anônimas. Variáveis com o mesmo nome aparecendo numa mesma cláusula são a mesma variável, ou seja, se uma ganha um valor, este valor passa imediatamente para as outras ocorrências, exceto para variáveis anônimas. (1) cada ocorrência delas indica uma variável diferente, mesmo dentro de uma mesma cláusula, e (2) ao serem usadas numa pergunta, seus valores não são impressos nas respostas. Variáveis anônimas são usadas quando queremos que unifiquem com qualquer termo mas não nos interessa com qual valor serão instanciadas.

Por fim, as estruturas são termos mais complexos formados por um funtor seguido de componentes separados por vírgula e colocadas entre parênteses. Por exemplo, para indicar um livro com seu título e autor podemos usar a estrutura abaixo:

```
livro(incidente_em_antares, verissimo)
```

Observe que os fatos de um banco de dados em Prolog são estruturas seguidas de um ponto final. Estruturas podem ser aninhadas, se quisermos sofisticar a indicação dos livros, colocando nome e sobrenome do autor para poder diferenciar entre vários autores com o mesmo sobrenome, podemos usar:

```
livro(incidente_em_antares, autor(erico, verissimo))
```

Estruturas podem ser argumentos de fatos no banco de dados:

```
pertence(ze, livro(incidente_em_antares, verissimo)).
```

REFERÊNCIAS BIBLIOGRÁFICAS

João Meidanis. **MC346 - Paradigmas de programação**: Prolog. 2014. Acesso em 1º de maio 2024. Disponível em: <<https://www.ic.unicamp.br/~meidanis/courses/mc346/2015s2/logica/apostila-prolog.pdf>>.

(João Meidanis, 2014)