

1)

A lógica proposicional permite caracterizar de forma rigorosa e precisa relacionamentos entre proposições. Consideramos uma extensão da lógica proposicional, visando torná-la mais expressiva, mais próxima da linguagem natural. Isso amplia as oportunidades de aplicação da lógica para inferir fatos a respeito dos sistemas que correspondam a sua semântica. Mas, para isso, será pago o preço de aumentar consideravelmente a complexidade da lógica.

Consideremos, por exemplo, as seguintes sentenças proposicionais:

$e_1 \rightarrow s_1$

$e_2 \rightarrow s_2$

$e_3 \rightarrow s_3$

Intuitivamente, os índices de valor 1 correspondem a Marcelo, os índices de valor 2 a Ana e os de índice 3 a Fábio. As proposições  $e_i$  denotam que “o indivíduo  $i$  pratica esportes” e as  $s_i$  denotam “o indivíduo  $i$  tem boa saúde”. Se os únicos indivíduos do domínio forem esses três, não precisa estender a lógica proposicional. Mas, se quisermos, por exemplo, escrever que “qualquer pessoa que pratica esporte tem boa saúde”? Precisaríamos de algo como: “ $e_i \rightarrow s_i$ ”, para qualquer valor de  $i \in I = \{1,2,3\}$ . Isso, entretanto, não pertence à linguagem proposicional.

A novidade é que as proposições representam agora propriedades relativas aos indivíduos correspondentes aos índices pertencentes a  $I$ . O nome usualmente adotado para essas propriedades é predicados, e por esse motivo, a lógica proposicional estendida dessa forma é chamada lógica de predicados ou lógica de primeira ordem (LPO).

2)

Um modelo (ou assinatura) para a linguagem será formado por um conjunto não vazio (chamado domínio ou universo), uma operação  $n$ -ária para cada símbolo funcional  $n$ -ário da linguagem, uma relação  $n$ -ária para cada símbolo relacional  $n$ -ário e um elemento do domínio para cada constante da linguagem.

3)

RELACIONAL:  $\text{gosta}(x, y)$ :  $x$  gosta de  $y$

$\forall x \exists y (\text{gosta}(x, y) \wedge \neg(x=y))$

RELACIONAL:  $\text{gosta}(x, y)$ :  $x$  gosta de  $y$

$\exists y \forall x (\text{gosta}(x, y) \wedge \neg(x=y))$

RELACIONAL:  $\text{frequenta}(x, y)$ :  $x$  frequenta  $y$

$\text{frequenta}(\text{João}, \text{IA}) \vee \text{frequenta}(\text{João}, \text{PE})$

RELACIONAL: frequenta(x, y): x frequenta y  
 $((\text{frequenta}(\text{João}, \text{IA}) \wedge \neg(\text{frequenta}(\text{João}, \text{PE}))) \vee$   
 $((\text{frequenta}(\text{João}, \text{PE}) \wedge \neg(\text{frequenta}(\text{João}, \text{IA})))$

RELACIONAL: irmãDe(x, y): x é irmã de y  
 $\forall x, y (\text{irmãDe}(\text{Ana}, x) \wedge \text{irmãDe}(\text{Ana}, y) \rightarrow (x=y))$

RELACIONAL: irmãDe(x, y): x é irmã de y  
 $\exists y (\text{irmãDe}(\text{Ana}, x) \wedge \forall y (\text{irmãDe}(\text{Ana}, y) \rightarrow (x=y)))$

RELACIONAL: irmãDe(x, y): x é irmã de y  
 $\exists x, y (\text{irmãDe}(\text{Ana}, x) \wedge \text{irmãDe}(\text{Ana}, y) \wedge \neg(x=y))$

4)

RELACIONAIS: pessoa(x): x é uma pessoa  
professor(x): x é um professor  
 $\forall x (\text{professor}(x) \rightarrow \text{pessoa}(x))$

RELACIONAIS: professor(x): x é um professor  
profTitular(x): x é professor titular  
 $\forall x (\text{profTitular}(x) \rightarrow \text{professor}(x))$

RELACIONAIS: departamento(x): x é um departamento  
organização(x): x é uma organização  
 $\forall x (\text{departamento}(x) \rightarrow \text{organização}(x))$

RELACIONAIS: pessoa(x): x é uma pessoa  
matriculada(x): x está matriculada  
estudante(x): x é um estudante  
 $\forall x (\text{pessoa}(x) \wedge \text{matriculada}(x) \rightarrow \text{estudante}(x))$

RELACIONAIS: departamento(x): x é um departamento  
chefia(x, y): x chefia y  
profTitular(x): x é professor titular  
 $\forall d (\text{departamento}(d) \wedge \exists y (\text{chefia}(y, t) \wedge \text{profTitular}(y)))$

5)

Prolog (Programação Lógica) é uma linguagem de programação que se enquadra no paradigma de Programação em Lógica Matemática. É uma linguagem de uso geral que é especialmente associada com a inteligência artificial e linguística computacional. Consiste numa linguagem puramente lógica, que pode ser chamada de Prolog puro, e numa linguagem concreta, a qual acrescenta o Prolog puro com componentes extra-lógicos.

O Prolog é uma linguagem declarativa, significando que em vez de o programa estipular a maneira de chegar à solução, passo a passo, (como nas linguagens

procedimentais ou imperativas), limita-se a fornecer uma descrição do problema que se pretende computar. Usa uma coleção base de dados de fatos e de relações lógicas (regras) que exprimem o domínio relacional do problema a resolver.

Um programa pode rodar num modo interativo, a partir de consultas (queries) formuladas pelo usuário, usando a base de dados (os 'fatos') e as regras relacionais (essencialmente implicações lógicas: se.. então), e o mecanismo de unificação para produzir (por uma cadeia de deduções lógicas) a solução.

O Prolog é baseado num subconjunto do cálculo de predicados de primeira ordem, o que é definido por cláusulas de Horn. A execução de um programa em Prolog é efetivamente a prova de um teorema por resolução de primeira ordem. Alguns conceitos fundamentais são unificação, recursão, e backtracking.