

# UTILIDADE DO CÁLCULO DOS PREDICADOS NA ÁREA DE COMPUTAÇÃO

## PROGRAMAÇÃO LÓGICA

Usando o conhecimento do cálculo proposicional, analisamos a validade de um argumento a partir da análise das premissas ou hipóteses. Para isso, fazemos uso das regras de inferência. Esse processo também ocorre com o cálculo de predicados.

**Ou seja, considerando que as premissas são verdadeiras, concluimos que o resultado (conclusão) é verdadeiro.**

Na Computação, uma das primeiras linguagens concebidas para implementar tais procedimentos foi a linguagem de programação Prolog.

### Você sabia

A linguagem Prolog foi desenvolvida em 1972 por Alain Colmerauer e seus associados na Universidade de Marseille, na França, com o propósito inicial de traduzir linguagens naturais, e tem uma importância considerável na área de Inteligência Artificial. Podemos citar como uma característica o fato de ser uma linguagem declarativa, também chamada de linguagem descritiva. Isso significa que, dado um problema, ela faz uso do cálculo proposicional e do cálculo de predicados para indicar como o problema deve ser resolvido. Isso é feito por meio da interpretação de quais hipóteses são verdadeiras.

A linguagem Prolog é formada basicamente de **fatos** ou **informações** sobre objetos que desejamos analisar; regras de inferência e perguntas que podemos fazer sobre os objetos dados.

### Atenção

Para resolver um determinado problema, o programa Prolog recorre a uma base de dados formada por fatos e regras. A partir dessa base de dados, determina-se se uma conclusão tem valor lógico verdadeiro ou falso.

Agora vamos entender o que é um fato. É a partir dos fatos que os predicados são definidos, pois a estrutura de um fato é formada por um predicado e seus objetos ou argumentos. Veja:

---

predicado (objeto 1, objeto 2, ...).

---

Veja alguns exemplos:

### amigo(paulo, carlos)

Observe que definimos uma relação entre dois objetos, “paulo e carlos”.  
“Paulo é amigo de Carlos”

**mãe(maria, pedro)**

“Maria é mãe de Pedro”

**gosta(paula, pedro)**

“Paula gosta de Pedro”

**Atenção:** o predicado representa uma característica do objeto quando temos apenas 1 objeto.

## Exemplo

**estudiosa(maria).**

“Maria é estudiosa”

Usamos letras minúsculas para escrever os nomes dos predicados e dos objetos. Como podemos observar nos exemplos anteriores, o predicado vem na frente dos objetos, que são separados por vírgula.

Por convenção, temos o predicado( $x, y$ ), que indica “ $x$  predicado  $y$ ”. Assim, em `come(gato, peixe)`, temos: “Gato come peixe”.

Veja que trocar a ordem dos objetos pode alterar o valor lógico.

## Exemplo

- **`come(gato, peixe)`, temos: “Gato come peixe” – valor lógico verdadeiro.**
- **`come(peixe, gato)`, temos: “Peixe come gato” – valor lógico falso.**

Também podemos fazer perguntas. Na programação Prolog, uma pergunta tem a seguinte estrutura:

---

**? - possui(carlos, computador).**

**Estamos perguntando se “Carlos possui um computador”.**

---

O programa Prolog responde “sim” ou “não”.

## Atenção

Vale sinalizar que a linguagem Prolog não responde simplesmente “sim” ou “não”; ela vai muito além disso, pois, utilizando regras, unifica informações, analisa os fatos, encontra valores para as variáveis em questão que podem indicar o valor lógico da conclusão. A resposta “não” no Prolog indica que não foi possível chegar à conclusão ou provar o fato dado a partir das informações encontradas no banco de dados.

### Exemplo

Vamos considerar o seguinte banco de dados:

- engenheiro(luis). Significa: Luis é engenheiro.
- engenheiro(carlos). Significa: Carlos é engenheiro.
- carioca(luis). Significa: Luis é carioca.

Vamos fazer a seguinte pergunta:

**? – carioca(carlos).**

Significa: Carlos é carioca?

Note que não é possível afirmar que Carlos é carioca se considerarmos os fatos dados. Na pergunta também podemos fazer uso de variáveis que são denotadas por letras maiúsculas.

### Atenção

O Prolog examina os fatos na ordem em que estes se encontram no banco de dados e encontrando um fato que satisfaça a pergunta, ele atribui o valor encontrado à variável usada na pergunta em questão. Além disso, o Prolog também marca a posição no banco de dados em que a atribuição foi realizada.

Caso o usuário entre com o comando para que a busca seja realizada, o Prolog reiniciará a busca a partir da posição em que havia parado. No caso da pergunta anterior, o programa, então, responderia: X = carlos.

Outra pergunta:

**? – gosta(Mario, X).**

**X = paula**

### Comentário

Veja que mostramos até agora perguntas simples, mas à medida que essas perguntas ficam mais complexas, é necessário fazer uso dos conectivos lógicos e símbolos da aritmética, se necessários.

2. “Será que Luiza gosta de maratona e Carlos gosta de meia-maratona?”

Pergunta:

**? – gosta(luiza, maratona), gosta(carlos, meia-maratona)**

### Exemplo

Luiza gosta de todas as pessoas que gostam de corrida.

Podemos escrever:

Luiza gosta de X se X gosta de corrida.

As regras, na linguagem Prolog, possuem duas partes e elas são conectadas pelo símbolo “:-” que representa o “se”.

A linguagem de programação Prolog, bem como as linguagens da família Lisp, foram as primeiras linguagens de programação concebidas para a definição de sistemas lógicos.

Saiba mais

Lisp (do inglês List Processing) é uma família de linguagens formais matemáticas, concebida por John McCarthy em 1958. Sua estrutura de dados básica é a lista, que representa tanto os dados quanto os programas, sendo conhecida como uma linguagem funcional por ser um mecanismo formal para manipular funções matemáticas. Lisp possui várias implementações, como Common Lisp, Scheme, Clojure, e se tornou, juntamente com o Prolog, nos anos 1970 e 1980, uma das principais linguagens da comunidade de Inteligência Artificial.

A linguagem Prolog possui diversas aplicações, sendo as principais na área de Computação Simbólica:

- 1. Prova automática de teoremas e semântica;**
- 2. Solução de equações simbólicas;**
- 3. Bancos de dados relacionais;**
- 4. Processamento de linguagem natural;**
- 5. Sistemas especialistas;**
- 6. Planejamento automático de atividades;**
- 7. Aplicações em jogos (xadrez, damas, jogo da velha etc.);**  
**Compiladores etc.**

No próximo tópico, vamos nos concentrar em uma dessas aplicações, os chamados sistemas especialistas.

## SISTEMAS ESPECIALISTAS

Sistemas especialistas são uma aplicação importante do cálculo dos predicados. Foram criados por pesquisadores da área de Inteligência Artificial, com o objetivo de simular o raciocínio de um profissional especialista em alguma área de conhecimento, para uma tomada de decisão específica.

A base de dados desse sistema é formada por fatos e regras capazes de ajudar de forma eficiente o usuário na tomada de decisão e dar sugestões aos usuários. Isso ocorre por meio de regras que analisam informações e simulações do raciocínio do homem em uma determinada área de estudo. Com as simulações, surgem questionamentos com relação à tomada de decisão.

Esse tipo de sistema foi desenvolvido, por exemplo, com a finalidade de:

1. Simular diagnósticos na área médica, uma vez que se tem as informações fornecidas pelos pacientes;
2. Tomada de decisões gerenciais em fábricas;
3. Decisões baseadas em pesquisa de mercado;

#### 4. Declaração de imposto de renda.

##### **Atenção**

Note que, nesse sistema, as informações não são quantitativas, e sim qualitativas. É importante extrair o máximo de informações possíveis do especialista humano. Assim, temos tomadas de decisão mais eficientes.

Com relação a regras, elas podem ser simples ou complexas em que o número de operadores lógicos é maior. Por exemplo:

---

**Regra: “Se Carla não estudar, então ela não será aprovada na disciplina.”**

---

O sistema especialista possui fatos referentes ao problema estudado. Os usuários fornecem as informações e, a partir disso, o sistema procura por regras para unir as premissas às informações dadas.

Segundo Hayes-Roth (1983), os sistemas especialistas podem ser classificados nas seguintes categorias:

Interpretação	Inferem descrições de situações a partir da observação de fatos.
Predição	Deduzem consequências prováveis de determinadas situações.
Diagnóstico	Deduzem conclusões a partir de fatos observados.
Projeto	Configuram objetos sob dadas restrições.
Planejamento	Criam iniciativas para se atingir metas.
Monitoramento	Comparam fatos observados para detectar vulnerabilidades.
Depuração	Fornecem soluções incrementais para problemas complexos.
Conserto	Executam um plano para administrar uma solução prescrita.
Instrução	Diagnosticam, avaliam e reparam o comportamento do aluno.
Controle	Interpretam, preveem, reparam e monitoram comportamentos do sistema.

## PROVA DE CORREÇÃO

Outra utilização do cálculo de predicados diz respeito à correção de programas desenvolvidos com linguagens de programação tradicionais. A verificação do programa tem por finalidade garantir que o programa está correto, atendendo às especificações dadas pelo usuário.

## **Comentário**

Note que a verificação é diferente da validação, que busca garantir que os requisitos do cliente sejam atendidos e, portanto, vai além da correção de programas.

A verificação é realizada por meio da “prova de correção”, em que se usa a lógica formal com o intuito de provar que quaisquer variáveis de entrada que atendem às especificações de predicados e as variáveis de saída produzidas pela execução do programa satisfazem outras propriedades específicas.

Tal prova é baseada na verificação da correção de trechos de programa que incluam comandos presentes em toda linguagem de programação, tais como:

- 1. Comandos de atribuição de valores a variáveis.**
- 2. Comandos condicionais (*IF THEN ELSE*).**
- 3. Comandos com laços de repetição (*LOOP, FOR, REPEAT UNTIL, WHILE* etc.).**

Além da prova de correção, os testes de programas envolvem conjuntos de dados para testes. Eles mostram que os valores de saída são bons à medida que os valores de entrada são aceitáveis.