

Programação Lógica

Lógica para Computação

Professora: Simone Santos

Universidade Federal do Ceará
Campus de Crateús

6 de Outubro de 2020

Visão geral da programação lógica

- Linguagens usadas para programação lógica são chamadas de **linguagens declarativas**.
- Os programas escritos nelas consistem em **declarações** em vez de atribuições e sentenças de fluxo de controle.
- Essas declarações são sentenças, ou proposições, em linguagem simbólica.

Visão geral da programação lógica

- Uma das características essenciais das linguagens de programação lógica é a sua semântica, chamada de **semântica declarativa**.
- Essa semântica é mais simples do que a das linguagens imperativas.
- O significado de uma proposição em linguagens de programação lógica pode ser concisamente determinado a partir da própria sentença.
- Em uma linguagem imperativa, a semântica de uma sentença de atribuição requer o exame de declarações locais, conhecimento das regras de escopo da linguagem e outras questões.

Visão geral da programação lógica

- A programação em uma linguagem de programação lógica não é procedural.
- Os programas não descrevem exatamente como um resultado será computado, mas a forma do resultado.
- O sistema de computação pode, de alguma forma, determinar como o resultado será computado.

Visão geral da programação lógica

- O que é necessário é um meio conciso de fornecer ao computador tanto as informações relevantes quanto um método de inferência para computar os resultados desejados.
- O **cálculo de predicados** fornece a forma básica de comunicação com o computador, e a **resolução** fornece a técnica de inferência.

Cláusulas de Horn

- Um programa lógico expressa especificações para soluções de problemas com o uso de expressões em lógica matemática.
- Este estilo evoluiu a partir das necessidades dos pesquisadores em processamento de linguagem natural e na prova automática de teoremas.
- As linguagens de programação convencionais não são bem adequadas para essas necessidades.

Cláusulas de Horn

A lógica proposicional e de predicados proporcionam os fundamentos formais para a programação lógica. A cláusula de Horn é uma variante particular de lógica de predicados que está por trás da sintaxe da Prolog.

Cláusula de Horn

É uma sentença que tem uma cabeça h , que é um predicado, e um corpo (que pode ser opcional), que pode ser visto como uma lista de atributos p_1, p_2, \dots, p_n .

Cláusulas de Horn

As cláusulas de Horn são escritas no seguinte estilo:

$$h \leftarrow p_1, p_2, \dots, p_n$$

Isso significa que h é verdadeiro somente se p_1, p_2, \dots, p_n forem simultaneamente verdadeiros.

Resolução

Definição

É uma regra de inferência que permite às proposições inferidas serem computadas a partir de proposições dadas. O princípio da resolução é similar à ideia da transitividade em álgebra.

Quando aplicada às cláusulas de Horn, a resolução diz que se h é a cabeça de uma cláusula de Horn e ela corresponde a um dos termos de uma outra cláusula, então aquele termo pode ser substituído por h .

Resolução

Em outras palavras, se temos as cláusulas

$$h \leftarrow terms$$

$$t \leftarrow t_1, h, t_2$$

então podemos resolver a segunda cláusula para

$$t \leftarrow t_1, terms, t_2$$

Resolução

Por exemplo, considere as seguintes cláusulas:

$fala(Maria, portugues)$

$falacom(x, y) \leftarrow fala(x, l), fala(y, l), x \neq y$

A primeira é uma cláusula de Horn sem corpo, assim a cabeça é incondicionalmente verdade.

Resolução

$fala(Maria, portuguese)$

$falacom(x, y) \leftarrow fala(x, l), fala(y, l), x \neq y$

A resolução nos permite deduzir o seguinte:

$falacom(Maria, y) \leftarrow$

$fala(Maria, Portuguese), fala(y, Portuguese), Maria \neq y$

com a hipótese de que às variáveis x e y são atribuídos os valores “Maria” e “Portugues”. A resolução, portanto, ajuda a chegar a conclusões.

Resolução

Definição

A atribuição de valores a variáveis durante a resolução é chamada de **instanciação**.

Unificação

É um processo de correspondência de padrões que determina que instanciações, em particular, podem ser feitas a variáveis ao mesmo tempo em que faz uma série de resoluções simultâneas.

Programação lógica em Prolog

- **Prolog** é a linguagem principal usada em programação lógica.
- O desenvolvimento de Prolog foi baseado na **resolução** e na **unificação**.
- As aplicações da programação lógica estão espalhadas pelas áreas de processamento de linguagem natural, raciocínio automático e prova de teoremas, pesquisa em bases de dados e sistemas especialistas.

Elementos de um programa Prolog

Os programas Prolog são feitos a partir de **termos**, que podem ser constantes, variáveis ou estruturas.

Uma **constante** é um átomo ou um inteiro não-negativo.

Uma **variável** é uma série de letras que deve começar com uma letra maiúscula.

Uma **estrutura** é um predicado com zero ou mais argumentos.

Elementos de um programa Prolog

Algumas estruturas de Prolog

```
animal(zebra)  
fala(Quem, russo)  
np(X, Y)
```


Elementos de um programa Prolog

Fatos e regras Prolog são realizações da ideia formal das cláusulas de Horn.

Um **fato** é uma sentença correspondente a uma cláusula de Horn sem o lado direito seguida por um ponto.

Uma **regra** é uma sentença correspondente a uma cláusula de Horn com cabeça e corpo.

Fatos de um programa Prolog

Fatos

```
fala(allen, russo)
```

```
fala(bob, ingles)
```

```
fala(maria, russo)
```

```
fala(maria, ingles)
```

Regras de um programa Prolog

Regra

```
falacom(Pessoa1,Pessoa2) :-  
fala(Pessoa1,L), fala(Pessoa2,L), Pessoa1 \= Pessoa2
```

Regras são interpretadas como asserções “somente se”, com vírgula desempenhando o papel de operador lógico “and”.

Prolog

Um programa Prolog é uma sequência de fatos e regras.

```
fala(allen, russo).  
fala(bob, ingles).  
fala(maria, russo).  
fala(maria, ingles).  
falacom(Pessoa1, Pessoa2) :-  
fala(Pessoa1, L), fala(Pessoa2, L), Pessoa1 \= Pessoa2.
```

Prolog

- Uma regra Prolog **tem sucesso** quando há instâncias de suas variáveis para as quais todos os termos à direita do operador :- simultaneamente tem sucesso para aquelas atribuições.
- Caso contrário, dizemos que a regra **falhou**.
- Um fato sempre tem sucesso, isto é, ele é universalmente verdadeiro.

Prolog

Por exemplo, a regra no programa dado tem sucesso para as instâncias:

```
Pessoa1 = allen  
Pessoa2 = maria
```

Pois há uma instância da variável *L* para russo para qual os três predicados da regras tem sucesso simultaneamente.

```
fala(allen, L)  
fala(maria, L)  
allen \= maria
```

Prolog

Para fazer um exercício com um programa Prolog, podemos escrever *queries* (consultas) no console de Prolog.

```
“Quem fala russo?”
```

```
?- fala(Quem, russo).
```

Em resposta, o programa tenta atender a uma query procurando um fato ou uma série de **aplicações fato/regra** que responderá à query, ou seja, uma atribuição de valores para as variáveis na query que faz um fato ou uma regra ter êxito, no sentido da **correspondência de padrões**.

Prolog

Para fazer um exercício com um programa Prolog, podemos escrever *queries* (consultas) no console de Prolog.

```
"bob fala com allen?"  
?- falacom(bob, allen).
```