Paradigma Lógico

Introdução

ERICK GALANI MAZIERO
erick.maziero@ufla.br

Departamento de Ciências da Computação
Universidade Federal da Lavras

Introdução

- Linguagens de **Programação Lógica** são baseadas na expressão de programas em uma **forma de lógica simbólica** e usa **processo de inferência lógico** para produzir os resultados

- Programas lógicos são declarativos, ao invés de procedurais
  - Indicam os resultados desejados
  - Não os passos para produzir os resultados

O que é lógica

- Já parou para pensar o que significa lógica?
- Você é um ser lógico?
- O dicionário de Oxford define lógica da seguinte forma:
  - *parte da filosofia que trata das formas do pensamento em geral (dedução, indução, hipótese, inferência etc.) e das operações intelectuais que visam à determinação do que é verdadeiro ou não.*

- Então, pode ser uma matéria tão abstrata e subjetiva, precisamos definir uma lógica mais formal, para tratamento computacional

Cálculo de predicados

- Antes de tratar sobre programação lógica, temos de definir a lógica formal, que é a base para as linguagens lógicas

Cálculo de predicados

- Proposição:
  - Sentença lógica que pode ser verdadeira ou falsa (booleana)
  - Consiste em uma forma de elencar objetos e relacionamentos entre esses objetos.
- Lógica formal:
  - Método para descrever proposições
  - Permitir que essas proposições, formalmente descritas, possam ser verificadas quanto à sua validade

Lógica simbólica

- Pode ser utilizada para 3 necessidades da lógica formal:
  1. Expressar proposições
  2. Relacionar proposições
  3. Descrever novas proposições a partir da inferência com base em outras proposições válidas
- Lógica Formal está bem ligada à matemática
  - Matemática pode ser pensada em termos lógicos
  - Por exemplo, os axiomas fundamentais da teoria dos números e conjuntos
  - Teoremas e suas provas

Cálculo de predicados de primeira ordem

- A forma da lógica simbólica usada para o paradigma lógico é chamado de cálculo de predicados de primeira ordem

Vamos, então fazer algumas definições

Proposição

- É uma sentença lógica, booleana
- Objetos em proposições:
  - Termos simples
    - Constantes
      - Representa um objeto
    - Variáveis
      - Pode representar objetos diferentes, em momentos diferentes
- Proposições simples ou atômicas
  - Consistem em termos compostos
    - Aparência de uma notação de função matemática

Proposição

```
pessoa(adinalbertil)
pessoa(moanijelica)
homem(adinalbertil)
mulher(moanijelica)
curte(adinalbertil, esportes)
curte(moanijelica, filmes)
```

- pessoa: **functor**
- adinalbertil: lista de **parâmetros**
  - 1-tupla
- adinalbertil, moanijelica: lista de **parâmetros**
  - 2-tupla
- adinalbertil, moanijelica, pessoa, homem, mulher, curte: **constantes**
  - Podem significar o que quisermos!!!
  - Embora sempre utilizemos palavras com significados de acordo com nosso senso comum

Proposição

- Fatos
  - Definidos explicitamente como verdadeiros
- Consultas
  - Realizam busca na base de fatos para validar busca

Proposição composta

- Duas ou mais proposições atômicas, conectadas por conectores ou operadores lógicos

Nome	*Símbolo*	*Exemplo*	*Significado*
negação	$\neg$	$\neg a$	não *a*
conjunção	$\cap$	$a \cap b$	*a* e *b*
disjunção	$\cup$	$a \cup b$	*a* ou *b*
equivalência	$\equiv$	$a \equiv b$	*a* é equivalente a *b*
implicação	$\supset$	$a \supset b$	*a* implica em *b*
	$\subset$	$a \subset b$	*b* implica em *a*

Proposição composta

- Duas ou mais proposições atômicas, conectadas por conectores ou operadores lógicos

$$a \cap b \supset c$$
$$a \cap \neg b \supset d$$

- O operador $\neg$ tem a precedência mais alta
- Os operadores $\cap$, $\cup$ e $\equiv$ têm precedência mais alta que $\supset$ ou $\subset$

Proposição composta: quantificadores

- Variáveis aparecem em proposições introduzidas por símbolos especiais: quantificadores
- Têm precedência mais alta que os operadores lógicos
- O cálculo de predicados inclui dois quantificadores:

Nome	*Exemplo*	*Significado*
universal	$\forall X.P$	Para todo X, P é verdadeiro.
existencial	$\exists X.P$	Existe um valor de X tal que P é verdadeiro.

X é uma variável e P e uma proposição

Proposição composta: quantificadores

- Exemplo

$$\forall X.\left(homem(X) \supset pessoa(X)\right)$$

ou

$$\exists X.\left(mae(maria, X) \cap homem(X)\right)$$

O que significam essas proposições?

Forma Clausal

- Para fins de programação lógica, quanto mais simples for a notação, melhor: evita redundância!
  - existem diversas maneiras de definir proposições com o mesmo significado!
- Então, define-se uma forma padrão, chamada de **Forma Clausal**
  - Todas proposições podem ser escritas em **forma clausal**, sem perder generalidade!

Forma Clausal

$$B_1 \cup B_2 \cup \ldots \cup B_n \subset A_1 \cap A_2 \cap \ldots \cap A_m$$

- A’s e B’s são termos
- Se todos os A’s são verdadeiros, então pelos menos um B é verdadeiro
- Quantificadores existenciais não são necessários
- Quantificadores universais são implícitos no uso de variáveis, nas proposições atômicas
- Necessita-se apenas das conjunções e disjunções

Forma Clausal

$$B_1 \cup B_2 \cup \ldots \cup B_n \subset A_1 \cap A_2 \cap \ldots \cap A_m$$

- Disjunções no lado esquerdo
  - Consequente
- Conjunções no lado direito
  - Antecedente

- Todas proposições de cálculo de predicados pode ser algoritmamente convertida para a forma clausal.
  - Nilsson (1971)

Forma Clausal

$gosta(bob, truta) \subset gosta(bob, peixe) \cap \text{peixe}(\text{truta})$

Bob Gosta de Truta, pois Bob gosta de peixe e truta é um peixe

$pai(louis, al) \cup pai\ (loius, violet)$
$\subset pai(al, bob) \cap mae(violet, bob) \cap avo(louis, bob)$

Interprete essa proposição

Cálculo de Predicados

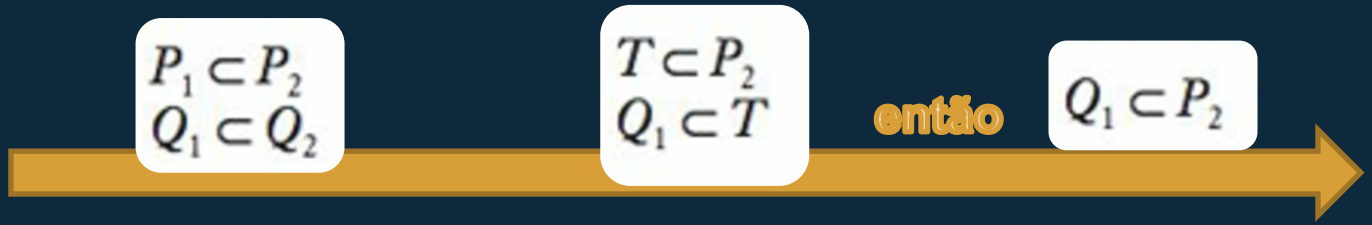
- É método para expressas coleções de proposições
- A partir dessas coleções, pode-se tentar determinar se um fato pode ser inferido (é válido)
- Isso é muito parecido com o trabalho de matemáticos que, a partir de axiomas e teoremas conhecidos, tenta provar novos teoremas

Cálculo de Predicados

- Nas décadas de 50 e 60, primeiros anos da Ciência da Computação, houve grande interesse em automatizar a prova de teoremas
  - Princípio da resolução, por Alan Robinson (1965), na Universidade de Syracuse, é um exemplo

Resolução

- É uma regra de inferência que permite às proposições inferidas serem computadas a partir de proposições dadas
- Método que pode ser aplicado na prova automática de teoremas
- Foi desenvolvida para ser aplicada em Forma Clausal



Resolução

- Outro exemplo

$$mais\_velho(joanne, jake) \subset mae(joanne, jake)$$
$$mais\_sabio(joanne, jake) \subset mais\_velho(joanne, jake)$$

O que pode se inferir das duas proposições???

Resolução

- Outro exemplo

$$mais\_velho(joanne, jake) \subset mae(joanne, jake)$$
$$mais\_sabio(joanne, jake) \subset mais\_velho(joanne, jake)$$

$mais\_sabio(joanne, jake) \subset mae(joanne, jake)$

Resolução: ’algoritmo’

- Os membros dos lados esquerdos são unidos por um **E** para fazer o novo lado esquerdo da nova proposição
- Faz-se o mesmo para o lado direito
- Qualquer termo que aparaça nos dois lados da nova proposição é removido dos dois lados

Resolução: ’algoritmo’

$$pai(bob, jake) \cup mae(bob, jake) \subset pais(bob, jake)$$
$$avo(bob, fred) \subset pai(bob, jake) \cap pai(jake, fred)$$

$$mae(bob, jake) \cup avo(bob, fred)$$
$$\subset pais(bob, jake) \cap pai(jake, fred)$$

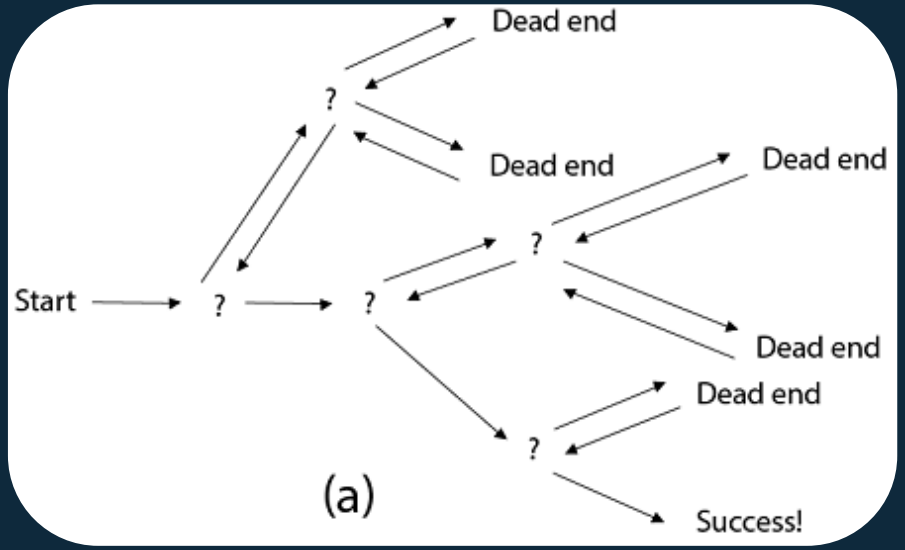
Resolução: ’algoritmo’

- A presença de variáveis nas proposições requer um processo mais complexo
  - Processo de determinar valores úteis para as variáveis, chamado de **unificação**
  - Atribuição temporária de valores a variáveis para permitir a unificação é chamada de **instanciação**

Instancia com um valor, falha, volta, instancia com outro valor

Resolução: backtracking

- A volta, é chamada de rastreamento para trás: **Backtracking**



Resolução: backtracking

- Retornaremos ao tópico quando falarmos de Prolog

- **Agora vamos definir um tipo específico de proposição: as cláusulas de Horn.**
- **Para isso, analisemos a prova de teoremas**

Prova de Teoremas

- Utilizamos a seguinte abordagem de prova por contradição
  - primeiro, nega-se o teorema a ser provado
  - depois, usa-se resolução
    - Se obtiver sucesso, o teorema não é verdadeiro
- As proposições originais são comumentes chamadas de hipóteses
- A negação do teorema é chamada de objetivo
- O tempo para aplicar a resolução pode ser muito grande, inviabilizando a aplicação dessa técnica, principalmente quando o conjunto de proposições é muito numeroso

Prova de Teoremas

- A prova de teoremas é a base para a programação lógica
  - Lista de fatos e relacionamentos: hipóteses
  - Objetivo a ser inferido, a partir das hipóteses, usando resolução

- Utiliza-se um tipo especial de proposições são usadas:

  Cláusulas de Horn
  - Alfred Horn (1951)

Cláusulas de Horn

- As cláusulas de Horn são um tipo específico de proposição da forma clausal e ocorre em duas possíveis formas:
  - Única proposição atômica do lado esquerdo
  - Lado esquerdo vazio

- Lado esquerdo: cabeça (cláusulas de Horn com cabeça)
  - Definir relacionamentos
  - $gosta(bob, truta) \subset gosta(bob, peixe) \cap \text{peixe}(\text{truta})$
- Lado esquerdo vazio (cláusulas de Horn sem cabeça)
  - Definir fatos
  - $pai(bob, jake)$

Visão Geral da Programação Lógica

- Linguagens de programação lógicas são declarativas
  - As sentenças, ou proposições são em lógica simbólica
- Usa semântica declarativa
  - Que é mais simples que em linguagens imperativas
    - Porém, exige conhecimento do contexto do comando
  - O significado pode ser determinados a partir da própria proposição
  - Tem vantagem em cima das linguagens imperativas
    - Hogger, 1984

Visão Geral da Programação Lógica

- Em paradigma imperativo e funcional:
  - Programação procedural
    - Detalhes do ”**Como**” resolver
- Em paradigma lógico:
  - Define-se ”**O que**”
  - O cálculo de predicados: comunicação com o computador
  - A resolução: técnica de inferência

Visão Geral da Programação Lógica

- Considere a tarefa de ordenação:
  - **Imperativo ou funcional**: descreve-se cada passo para receber uma lista de valores ordenáveis e produzir lista ordenada
  - **Lógico**: descreve-se que a lista ordenada é alguma permutação da lista de entrada tal que para cada par de elementos adjacentes, um relacionamento (<) se mantém entre eles.

Visão Geral da Programação Lógica

- **Lógico**: descreve-se que a lista ordenada é alguma permutação da lista de entrada tal que para cada par de elementos adjacentes, um relacionamento (<) se mantém entre eles.
- Imagine uma lista como um vetor numa faixa de índices de $1..n$

Possível solução em paradigma lógico:

$$sort(old\_list, new\_list)$$
$$\subset permute(old\_list, new\_list) \cap \text{sorted(new\_list)}$$
$$sorted(list) \subset \forall j \; such \; that \; 1 \leq j < n, list(j)$$
$$\leq list(j+1)$$

Prolog

- **Nos próximos slides a linguagem Prolog será apresentada e diversos exemplos serão dados na linguagem**

Referência Bibliográfica

Sebesta, R. W. (2011). *Conceitos de Linguagens de Programação*. 9 ed. Bookman.

Capítulo 16