
Programação em Lógica e Prolog

1. Introdução

Programa = Conjunto de Axiomas

Computação = Prova construtiva de teorema (pergunta)

- Hilbert (1925) - Programa de mecanização da matemática;
- Robison (1965) - Unificação e resolução;
- Kowalski (1972) - Interpretação procedural das cláusulas de Horn;
- Colmerauer (197?) - Implementação eficiente de Prolog.

2. Prolog Puro

Fatos

+

Regras

- Cláusulas de Horn;

Linguagem:

- Termos e fórmulas atômicas como em lógica de primeira ordem.

Regras

$A \leftarrow L1, L2, \dots, Ln$

- Interpretação: **A** se **L1** e **L2** e ... e **Ln**. Onde **A**, **L1**, **L2**, ..., **Ln** são átomos 'positivos'.
- Equivalente em lógica de primeira ordem a: $L1 \wedge L2 \wedge \dots \wedge Ln \rightarrow A$.
- Observação: em Prolog Puro (Cláusula de Horn) não há negação.
- **A** é dito cabeça ou consequente e **L1**, **L2**, ..., **Ln** é a cauda ou pré-condição da regra.

Exemplo:

$Alto(Joao) \leftarrow Jogador_Basq(Joao) , Homem(Joao)$

Fatos

$A \leftarrow$

- Interpretação: A é verdade sem nenhuma pré-condição.
- A é uma fórmula atômica 'positiva'.
- Equivalente na lógica clássica: fórmulas atômicas.

Exemplo:

Jogador_Basq(Joao) \leftarrow

Homem(Joao) \leftarrow

Exemplo:

Pai

Ivo	Pedro
Ivo	Ivone
Pedro	Rafael

Mãe

Vanda	Pedro
Vanda	Ivone
Ivone	Teresa

Homem

Ivo
Pedro
Rafael

Mulher

Vanda
Ivone
Teresa

Fatos:

Pai(Ivo,Pedro) ←
Pai(Ivo,Ivone) ←
Pai(Pedro,Rafael) ←

Mae(Vanda,Pedro) ←
Mae(Vanda,Ivone) ←
Mae(Ivone,Teresa) ←

Homem(Ivo) ←
Homem(Pedro) ←
Homem(Rafael) ←

Mulher(Vanda) ←
Mulher(Ivone) ←
Mulher(Teresa) ←

Quantificação Universal

- Variáveis nos fatos e nas regras são implicitamente quantificadas universalmente.
- $A \leftarrow L1, L2, \dots, Ln$
- Equivalente em lógica de primeira ordem a:

$\forall X1 \forall X2 \dots \forall Xm \ L1 \wedge L2 \wedge \dots \wedge Ln \rightarrow A$. Onde $X1, X2, \dots, Xm$ são todas as variáveis livres ocorrendo em $L1, L2, \dots, Ln$ e A .

Exemplo: Definir Filho(X,Y) = X é filho de Y.

1. Filho(X,Y) \leftarrow Homem(X) , Pai(Y,X).
2. Filho(X,Y) \leftarrow Homem(X) , Mae(Y,X).

Leitura: Para todo X e Y, X é filho de Y se X é homem e Y é pai **ou** mae de X.

- Algumas vezes é útil usar quantificação implícita para fatos:

Exemplo: $\text{Ancestral}(X,Y) = X \text{ é ancestral de } Y$.

Suponha que queremos representar que Adão é ancestral de todos.

$\text{Ancestral}(\text{Adão}, Z) \leftarrow$

Significa: para todo Z Adão é ancestral de Z .

Perguntas

- Perguntas são conjunções de fórmulas atômicas sempre implicitamente quantificadas existencialmente.
- Notação: $\leftarrow L1, L2, \dots, Ln$.
- Onde $L1, L2, \dots, Ln$ são fórmulas atômicas 'positivas'.

Exemplo:

Filho(X, José), Filho(X, Maria)?

- Significa existe algum X no universo de discurso que é filho de José e filho de Maria?

Resolução LSD

- Resolução Linear com função de Seleção para cláusulas Definidas.
- **Cláusula Definida:** qualquer fato ou regra.
- **Cláusula de Horn:** qualquer fato ou regra ou pergunta.
- $A \leftarrow L_1, L_2, \dots, L_n$ representa $\forall X_1 \forall X_2 \dots \forall X_m L_1 \wedge L_2 \wedge \dots \wedge L_n \rightarrow A$.
- Convertendo em cláusulas temos: $\sim L_1 \vee \sim L_2 \vee \dots \vee \sim L_n \vee A$.
- Possui um único literal positivo, i.e., a cabeça.
- $\leftarrow L_1, L_2, \dots, L_n$ representa $\exists X_1 \exists X_2 \dots \exists X_m L_1 \wedge L_2 \wedge \dots \wedge L_n$.
- Convertendo em cláusulas temos: $\sim L_1 \vee \sim L_2 \vee \dots \vee \sim L_n$.
- Possui somente literais negativos.

Exemplo:

1. $p \leftarrow q, r$

2. $q \leftarrow s$

3. $r \leftarrow t$

4. $s \leftarrow$

5. $t \leftarrow$

6. $\leftarrow p$

7. $\leftarrow q, r$

8. $\leftarrow s, r$

9. $\leftarrow r$

10. $\leftarrow t$

11. \leftarrow

1. $p \vee \sim q \vee \sim r$

2. $q \vee \sim s$

3. $r \vee \sim t$

4. s

5. t

6. $\sim p$

7. $\sim q \vee \sim r$

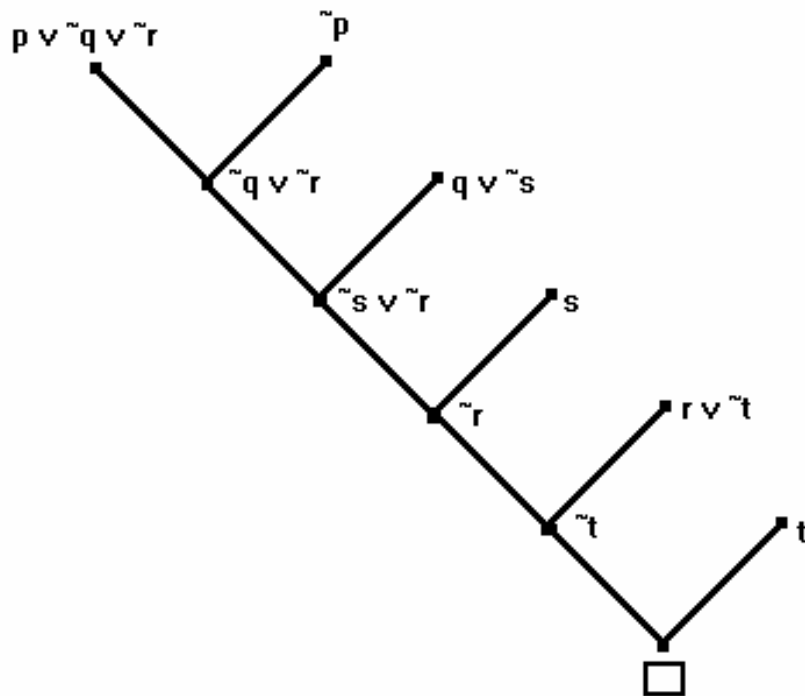
8. $\sim s \vee \sim r$

9. $\sim r$

10. $\sim t$

11. \uparrow

- A estratégia é linear de entrada: Cada passo é o resolvente do anterior com alguma cláusula de entrada.



A função de seleção escolhe o literal a ser resolvido a cada passo. No nosso caso o mais à esquerda.

Interpretação Procedural

$A \leftarrow L1, L2, \dots, Ln$

- Interpretação 1: para resolver **A** é preciso resolver **L1** e resolver **L2** e ... e resolver **Ln**.
- Interpretação 2: Chamada de procedimento. O procedimento **A** chama o procedimento **L1** e depois o **L2** e ... e por último o **Ln**.

$A \leftarrow L1, L2, \dots, Ln$

Procedure A

call L1

call L2

•

•

•

call Ln

Exemplo:

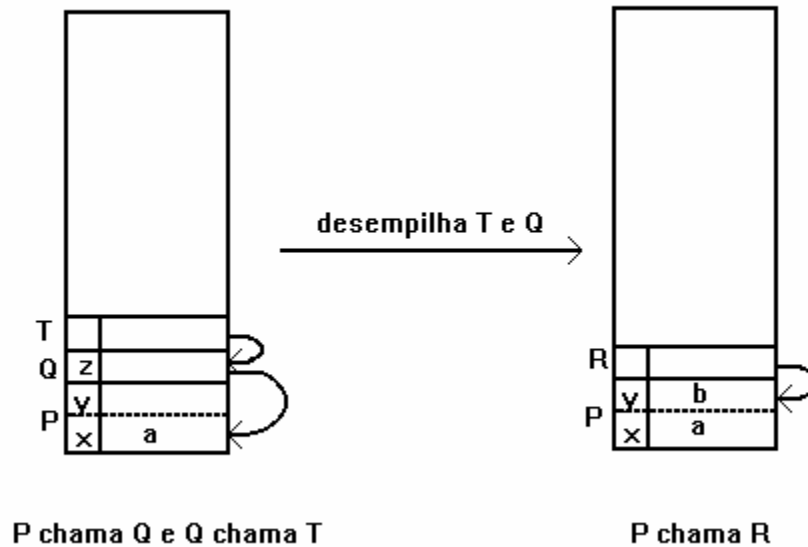
$P(X,Y) \leftarrow Q(X), R(Y)$

$Q(Z) \leftarrow T(Z)$

$T(a) \leftarrow$

$R(b) \leftarrow$

$\leftarrow P(X,Y)$



Interpretador para Prolog Puro

Entrada: Conjunto de cláusulas definidas P e uma pergunta R .

Saída: Uma substituição θ aplicada a R , tal que $R\theta$ é derivável de P , ou *falha* se uma substituição não puder ser achada.

Algoritmo: Inicialize a pergunta R como o primeiro resolvente
enquanto o resolvente não está vazio faça
 Escolha um literal A do resolvente (ex. o mais a esquerda) e
 uma regra $A' \leftarrow L_1, L_2, \dots, L_n$ onde A' é unificável com A por um
 unificador θ . Se não existe tal regra e tal θ sai do enquanto.
 Remova A e adicione L_1, L_2, \dots, L_n ao resolvente.
 Aplique θ ao novo resolvente e R .
Se o resolvente está vazio então retorna R .
 senão retorna *falha*.

Exemplo: Família

1. $\text{Filho}(X,Y) \leftarrow \text{Homem}(X) , \text{Pai}(Y,X).$
2. $\text{Filho}(X,Y) \leftarrow \text{Homem}(X) , \text{Mae}(Y,X).$
3. $\text{Filha}(X,Y) \leftarrow \text{Mulher}(X) , \text{Pai}(Y,X).$
4. $\text{Filha}(X,Y) \leftarrow \text{Mulher}(X) , \text{Mae}(Y,X).$

Avô e Avó.

5. $\text{Avô}(X,Y) \leftarrow \text{Pai}(X,Z) , \text{Pai}(Z,Y).$
6. $\text{Avô}(X,Y) \leftarrow \text{Pai}(X,Z) , \text{Mae}(Z,Y).$
7. $\text{Avó}(X,Y) \leftarrow \text{Mae}(X,Z) , \text{Pai}(Z,Y).$
8. $\text{Avó}(X,Y) \leftarrow \text{Mae}(X,Z) , \text{Mae}(Z,Y).$

Pais e Avós.

9. $\text{Pais}(X,Y) \leftarrow \text{Pai}(X,Y).$
10. $\text{Pais}(X,Y) \leftarrow \text{Mae}(X,Y).$
11. $\text{Avós}(X,Y) \leftarrow \text{Avô}(X,Y).$
11. $\text{Avós}(X,Y) \leftarrow \text{Avó}(X,Y).$

Ou é pai ou é mãe.

Ou é avô ou é avó.

Exercício 1: faça as resoluções LSD para as seguintes perguntas:

1. $\text{Avô}(\text{Ivo}, \text{Teresa})$
2. $\text{Avô}(X, \text{Rafael})$
3. $\text{Avós}(X, \text{Rafael})$, $\text{Avós}(X, \text{Teresa})$

Para a pergunta 3 esboçe a pilha de execução.

Exercício 2: represente as seguintes relações:

1. Se um casal procriou, i.e., teve filhos.
2. Irmão, Irmãos, primos e tios.

X é irmão de Y.

$\text{Irmão}(X,Y) \leftarrow \text{Pais}(Z,X) , \text{Pais}(Z,Y) , \text{Homem}(X).$

Pergunta: $\leftarrow \text{Irmão}(X,X).$

Resposta será positiva.

$\text{Irmão}(X,Y) \leftarrow \text{Pais}(Z,X) , \text{Pais}(Z,Y) , \text{Homem}(X) , X \neq Y.$

Relação de diferente pode ser feito de duas maneiras:

1) Dar explicitamente quem é diferente de quem:

Ivo \neq Rafael \leftarrow	Maria \neq Rafael \leftarrow	Vanda \neq Maria \leftarrow
Ivo \neq Teresa \leftarrow	Teresa \neq Rafael \leftarrow	Ivo \neq Maria $\leftarrow \dots$

2) Definir uma relação entre termos: $\text{termo1} \neq \text{termo2}$ é verdade se termo1 e termo2 são diferentes e falso caso contrário.

Informação Estruturada

CURSOS							
nome	local			prof.	data		
	cidade	univer.	sala		dia	mes	ano

1) Linear

Curso(N,C,U,S,P,D,M,A) Curso(IA,Salvador,UFBA,15,Mario,12,11,93)

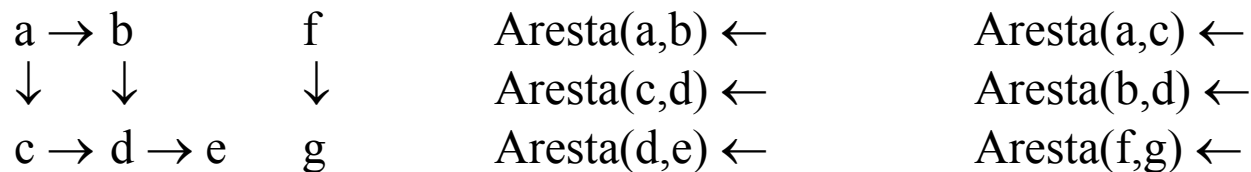
2) Estruturar usando funções: **record's**

Curso(nome, local(cidade,univer,sala), prof, data(dia, mes, ano))

Curso(IA, local(Salvador, UFBA, 15), Mario, data(12,11,93))

Regras Recursivas

- define uma relação em termos dela própria. Como se fosse um programa recursivo que chama a si próprio.
- Seja a relação que X é um ancestral de Y:
 1. $\text{Ancestral}(X,Y) \leftarrow \text{Pais}(X,Y).$
 2. $\text{Ancestral}(X,Y) \leftarrow \text{Pais}(X,Z) , \text{Ancestral}(Z,Y).$
- Conectividade num grafo: quem tem conexão com quem.



1. $\text{Conectado}(X,X) \leftarrow$
2. $\text{Conectado}(X,Y) \leftarrow \text{Aresta}(X,Z) , \text{Conectado}(Z,Y).$

- Uma definição recursiva sempre tem uma regra que chama a si própria para casos cada vez mais simples e sempre tem uma outra que é o caso base,i.e., o caso mais simples.
- Exercício: Seguir a execução do programa do grafo para a pergunta:

$\leftarrow \text{Conectado}(a,e)$

Ordem das Cláusulas

- De cima para baixo e da esquerda para a direita.
 - A ordem faz diferença na busca das soluções.
1. $\text{Ancestral}(X,Y) \leftarrow \text{Pais}(X,Y).$
 2. $\text{Ancestral}(X,Y) \leftarrow \text{Pais}(X,Z) , \text{Ancestral}(Z,Y).$
- Trocando 1. e 2. altera a ordem de computação das respostas.

Correção e Completude da Resolução LSD

- Seja P um conjunto de cláusulas definidas e R uma pergunta.

Correção: *toda resposta dada pelo algoritmo de resolução LSD é logicamente correta, i.e., se o algoritmo responde $R\theta$ é por que $R\theta$ é consequência lógica de P .*

Completude: *se existe resposta o algoritmo de resolução LSD acha, i.e., se existe uma resposta $R\theta$ que é consequência lógica de P então o algoritmo acha.*

3. Prolog com Negação

- A negação em Prolog está baseado num paradigma chamado de **negação por falha finita**.
- O método de resolução agora se chama LSD com negação por falha finita **LSDNF**.
- Um a cláusula pseudo-definida é uma cláusula da forma:

$A \leftarrow L1, L2, \dots, Ln$

Onde **A** é um literal positivo e **L1, L2, ..., Ln** são literais positivos ou negativos.

- Uma pseudo-pergunta é uma cláusula da forma:

$\leftarrow L1, L2, \dots, Ln$

Onde **L1, L2, ..., Ln** são literais positivos ou negativos.

- Literais negativos somente podem ocorrer do lado direito de \leftarrow . Nunca na cabeça, i.e., a cabeça é sempre positiva.
- Uma pergunta: $\leftarrow \text{not } R$ termina com sucesso se todas as tentativas de se provar R terminarem com falha.

Exemplo 1: $\leftarrow \text{not Pai(Paulo,Jorge)}$ vai terminar com sucesso pois este fato não pode ser obtido da nossa base de conhecimento.

Exemplo 2:

$\text{Orfão}(X) \leftarrow \text{not Pai}(X)$
 $\text{Orfão}(X) \leftarrow \text{not Mae}(X)$

Exemplo 3:

$\text{Estudante_solteiro}(X) \leftarrow \text{not casado}(X) , \text{estudante}(X).$

- Infelizmente resolução LSDNF não é correta e completa.

Exemplo:

Estudante_solteiro(X) \leftarrow not Casado(X) , Estudante(X).
Estudante(João)
Casado(José)

- Não possível provar que João é um estudante solteiro.
- Porém, trocando a ordem dos literais na primeira cláusula podemos provar que João é um estudante solteiro.

Estudante_solteiro(X) \leftarrow Estudante(X) , not Casado(X).

- Negação por falha finita é muito útil na prática, mas devemos ser bastante cuidadosos com o seu uso.
- Para saber se objetos são diferentes basta saber se eles não são ditos explicitamente que são iguais.