

Protótipo de um Interpretador para um Ambiente de Programação Lógica

Acadêmico: Wendy Krepsky
Orientador: Roberto Heinzle

FURB - Universidade Regional de Blumenau
Bacharelado em Ciências da Computação

Roteiro da Apresentação

- Introdução
- Programação de Computadores
- Inteligência Artificial
- Programação Lógica
- Ambiente Prolog
- Implementação do Protótipo
- Conclusão

Introdução

- Área
- Motivação
- Objetivos

Programação de Computadores

- O Computador é uma máquina genérica programável
- É programado em linguagem de máquina ou linguagem de programação
- Paradigmas:
 - Procedural Imperativo
 - Declarativo: funcional ou lógico

Inteligência Artificial (IA)

- Inteligência
- Inteligência Artificial
- Áreas de Aplicação
 - Processamento de Linguagem Natural (PLN)
 - Sistemas Especialistas (SE)

Programação Lógica (PL)

- Relações
- Classificação e hierarquização
- Lógica
- Programação em Lógica
 - Fatos e Regras
- Características
 - Lógica e Controle
- Áreas de Aplicação

Aspectos Matemáticos: Lógica das Proposições

■ Fórmulas Atômicas:

- A, B, PedroTemCarro

■ Símbolos:

- Conectivos: \wedge (e), \vee (ou), \Rightarrow (se e somente se), \neg (negação).

■ Regras de Sintaxe:

- Uma fórmula atômica é uma fbf
- Se α é uma fbf então $(\neg \alpha)$ é uma fbf
- Se α e β são fbfs, então $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $(\alpha \Rightarrow \beta)$ e $(\alpha \equiv \beta)$ são fbfs

Exemplos de fbfs:

$$(A \wedge B) \Rightarrow (\neg C)$$

$$A \Rightarrow (\neg B)$$

$$(A \vee B) \Rightarrow C$$

$$(A \Rightarrow B) \Rightarrow (\neg C \Rightarrow \neg D)$$

$$\neg \neg A$$

Aspectos Matemáticos: Lógicas das Proposições

■ Tabela Verdade

A	B	$A \wedge B$	$A \vee B$	$A \Rightarrow B$	$A \equiv B$	$\neg A$	$\neg\neg A$	$\neg A \vee B$
V	V	V	V	V	V	F	V	V
F	V	F	V	V	F	V	F	V
V	F	F	V	F	F	F	V	F
F	F	F	F	V	V	V	F	V

- Teorema da Dupla Negação: $(\neg(\neg A)) \equiv A$
- Teorema da Implicação: $(A \Rightarrow B) \equiv (\neg A) \vee B$

Aspectos Matemáticos: Lógica dos Predicados

■ Fórmulas Atômicas

- $\text{pai}(X, Y)$, $\text{homem}(X)$

■ Símbolos: (além de)

- Quantificadores: \forall (para todo), \exists (existe)

■ Regras de Sintaxe: (além das já citadas)

- se t_1, \dots, t_n são termos e p é um predicado que tem n argumentos, então $p(t_1, \dots, t_n)$ é uma fbf
- se V é uma variável e α é uma fbf, então $(\forall V \alpha)$ e $(\exists V \alpha)$ são fbfs

Exemplos de fbfs:

$\text{pai}(\text{joão}, \text{vitor})$

$\forall X \forall Y (\text{pai}(Z, X) \wedge \text{pai}(Z, Y) \Rightarrow \text{irmao}(X, Y))$

$\forall X (\text{humano}(X) \Rightarrow \text{mortal}(X))$

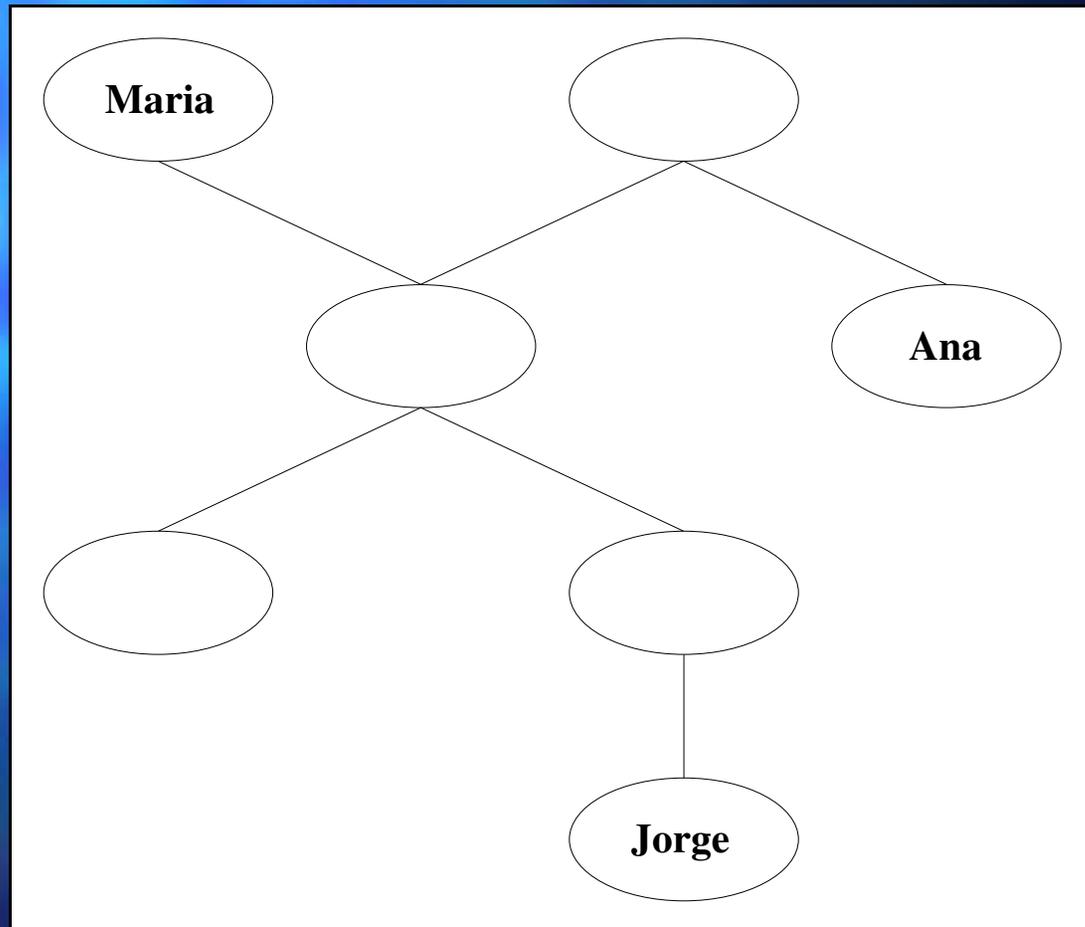
$\exists X \exists Y (\text{humano}(X) \wedge \text{humano}(Y) \wedge \text{ama}(X, Y))$

Ambiente Prolog

- Exemplo
- Fatos
- Regras
- Construções Recursivas

Prolog: Exemplo

- Família



Prolog: Fatos

```
progenitor(maria, josé).  
progenitor(joão, josé).  
progenitor(joão, ana).  
progenitor(josé, júlia).  
progenitor(josé, íris).  
progenitor(íris, jorge).
```

```
?- progenitor(josé, íris).  
sim  
?- progenitor(ana, jorge).  
não  
?- progenitor(luís, íris).  
não
```

```
?- progenitor(X, íris).  
X=josé  
?- progenitor(josé, X).  
X=júlia;  
X=íris;  
não
```

```
?- progenitor(X, Y).  
X=maria Y=josé;  
X=joão Y=josé;  
X=joão Y=ana;  
X=josé Y=júlia.
```

```
?- progenitor(X, Y), progenitor(Y, jorge).  
X=josé Y=íris.
```

Prolog: Regras

```
filho(Y, X) :-  
    progenitor(X, Y).  
avo(X, Z) :-  
    progenitor(X, Y),  
    progenitor(Y, Z).  
irmao(X, Y) :-  
    progenitor(Z, X),  
    progenitor(Z, Y).
```

$\forall X \forall Y (\text{progenitor}(X, Y) \Rightarrow \text{filho}(Y, X))$

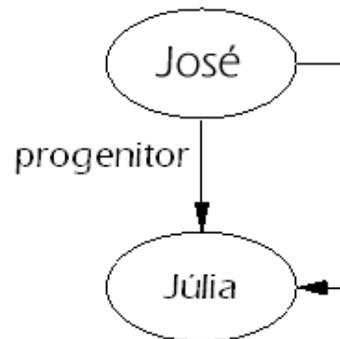
```
?- filho(X, josé).  
X=júlia;  
X=íris.  
?- avo(X, jorge).  
X=josé;  
?- irmao(X, íris).  
X=júlia;  
X=íris.
```

```
?- filho(jorge, X), irmao(X, Y), avo(joão, Y).  
X=íris Y=júlia.
```

Prolog: Construções Recursivas

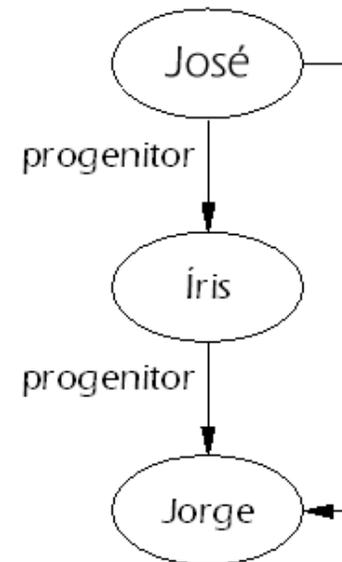
```
antepassado(X, Z) :-  
    progenitor(X, Z).  
antepassado(X, Z) :-  
    progenitor(X, Y),  
    antepassado(Y, Z).
```

```
?- antepassado(maria, julia).  
sim  
?- antepassado(X, jorge).  
X=íris;  
X=joão;  
X=maria.
```



(a)

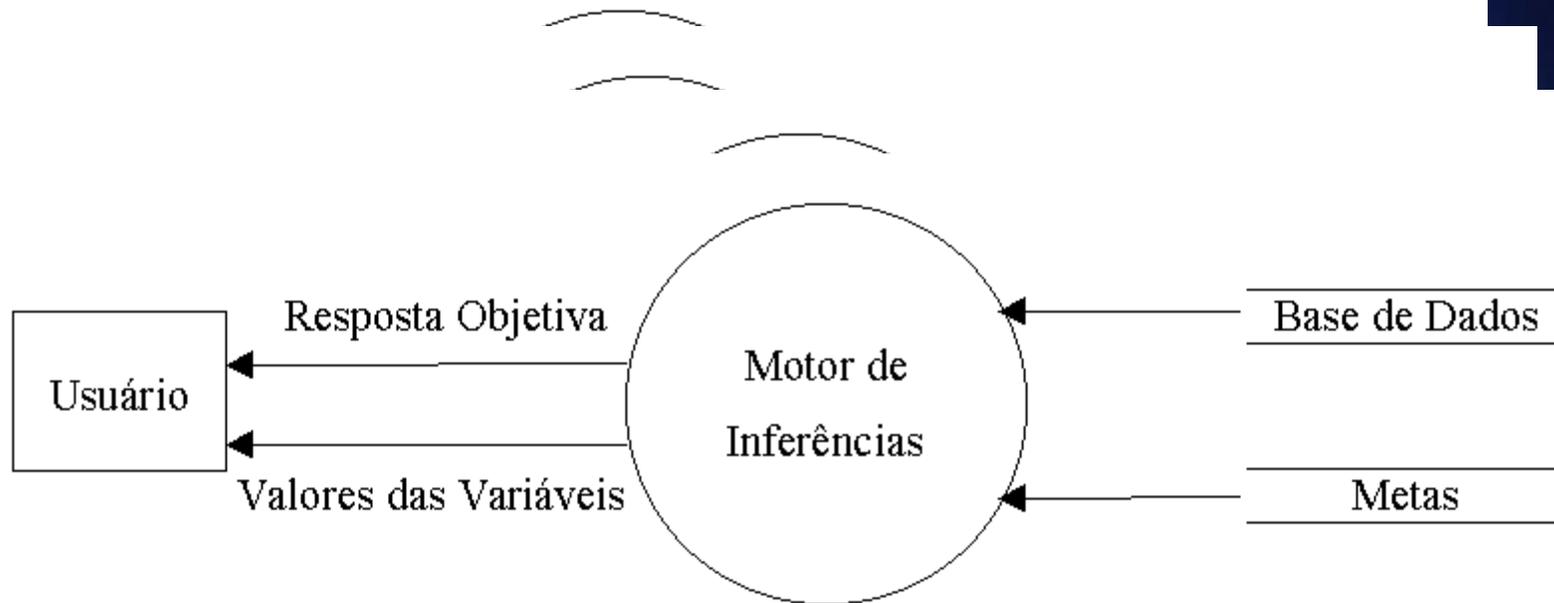
antepassado direto



(b)

antepassado indireto

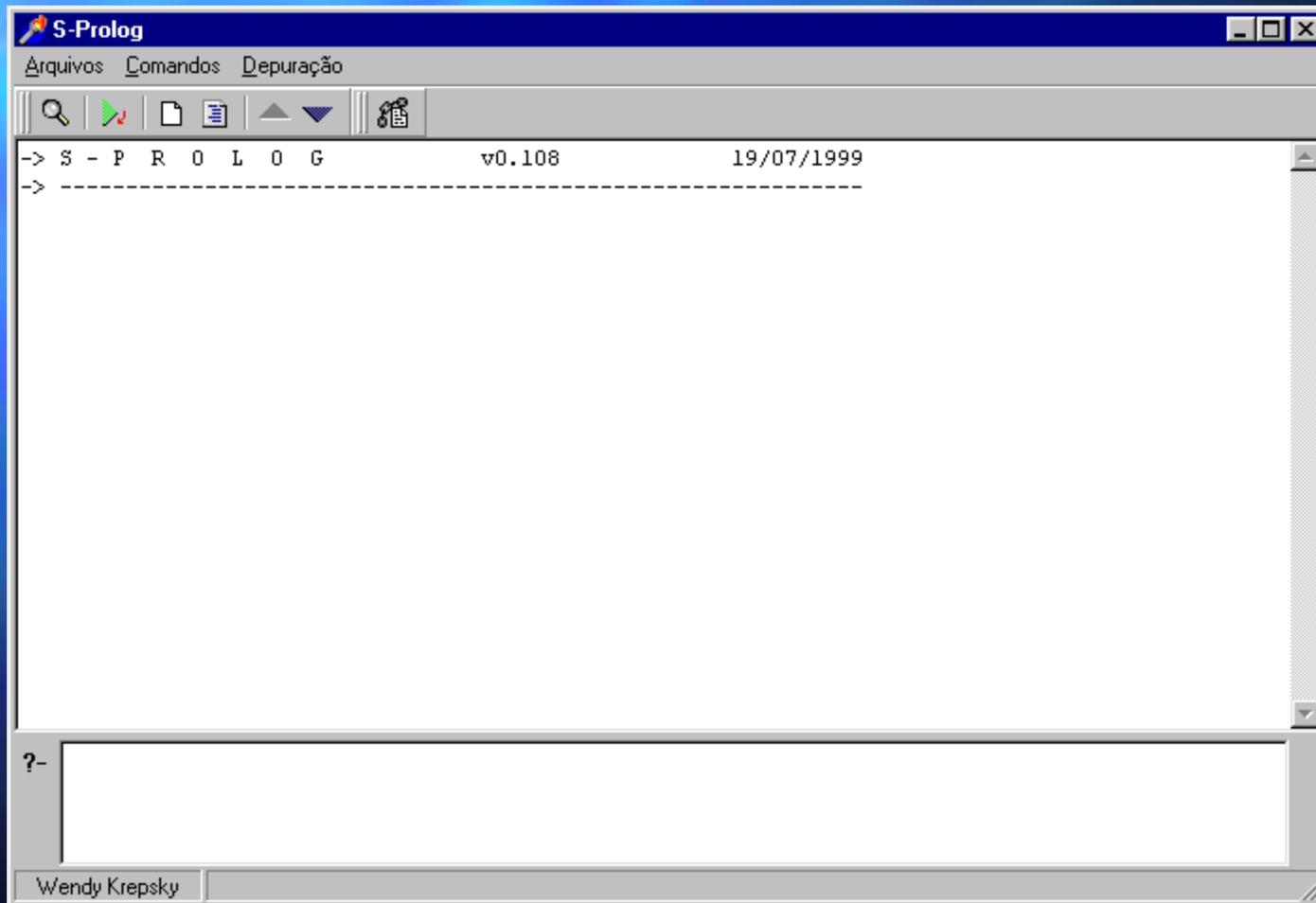
Implementação do Protótipo: Especificação



Implementação do Protótipo

- Metodologia Utilizada:
 - Prototipação
- Ferramenta Utilizada:
 - Delphi 4

Protótipo: Operacionalidade



Conclusões

- Estudo sobre Programação Lógica
- Implementação do Protótipo
 - Ferramenta
 - Metodologia

Extensões

■ Parte Teórica

- Estudo mais aprofundado sobre técnicas de resolução e unificação

■ Protótipo

- Implementar: unificação de cláusulas, operadores matemáticos, operador *cut*
- Mostrar todas as respostas válidas
- Dar suporte a manipulação de estruturas e listas