
UNIVASF
FUNDAÇÃO UNIVERSIDADE FEDERAL
DO VALE DO SÃO FRANCISCO

Lógica de primeira ordem

(Capítulo 8 - Russell)

Inteligência Artificial

Estrutura

1- Contextualização

2- Definições

3- Lista de exercício

4- Prolog

5- Regras em Prolog - Mundo Wumpus

6- Aplicação do Mundo Wumpus com Java e Prolog

O que não é possível expressar em Lógica Proposicional?

- Todo tricolor é um campeão. Roberto é tricolor. Logo Roberto é um campeão.
- A adição de dois números ímpares quaisquer é um número par.
- Acesso a esse recinto é permitido somente para as pessoas autorizadas ou conhecidas de pessoas autorizadas.

Por quê?

Ausências da Lógica Proposicional

Quantificadores

todo, qualquer, existe, alguns, nenhum, ...

Sempre estão ligados a variáveis

Objetos

Indivíduos do universo de discurso, sobre o qual quantificadores podem ser aplicados

Todo tricolor é um campeão. Roberto é tricolor.

Lógica de Predicados

Também chamada de

- Lógica de 1ª. Ordem
- FOL (First-Order Logic)

Extensão da Lógica Proposicional

- Novos conectivos (quantificadores)
- Novos símbolos para funções, variáveis, predicados, etc

Alfabeto

O alfabeto da Lógica de Predicados é constituído por:

símbolos de pontuação: (,);

símbolo de verdade: false;

um conjunto enumerável de símbolos para variáveis:

x, y, z, w, x_1,y_1,... ;

Alfabeto

Um conjunto enumerável de símbolos para funções:

$f, g, h, f_1, g_1, h_1, f_2, g_2, \ldots$;

Um conjunto enumerável de símbolos para predicados:

$p, q, r, p_1, q_1, r_1, p_2, q_2, \ldots$;

Conectivos:

$\neg, \vee, \forall, \exists$.

Associado a cada símbolo para função ou predicado, temos um número inteiro não-negativo k.

Esse número indica a aridade, ou seja, o número de argumentos da função ou predicado.

Alfabeto

- Constantes
- Variáveis
- Funções
- Predicados
- Conectivos

Constantes

Dão nomes a coisas particulares

Exemplo: Rosalvo, Brasil, Petrolina

Variáveis

Análogo a linguagens de programação.

Exemplo: x, y, z

Funções

Semelhante a função em programação, recebe um ou mais argumentos e produz como resposta um elemento do domínio como um número ou um objeto.

Exemplo: soma(x, y)

Predicados

Semelhante a uma função em programação com resposta booleana, a resposta será sempre verdadeiro ou falso. Utilizado para representar relações.

Exemplo: irmao(x, y), pai(x,y), vizinho(x,y)

Conectivos

Quantificadores

- Universal: $\forall$ (para todo …)
- Existencial: $\exists$ (existe …)

Os conectivos $\rightarrow$, $\leftrightarrow$ e ^ são definidos em função do conjunto completo $\{\neg, v\}$

E as fórmulas da lógica de predicados?

Para definir as regras para formação das fórmulas bem formadas é preciso estabelecer dois conceitos importantes:

-Átomos

- Termos

Tipos de perguntas (consultas)

"A capital de Pernambuco é Petrolina?"

Deve retornar um símbolo de verdade

Sentenças que representam símbolos de verdade, em Lógica de Predicados, são chamados de átomos

"Qual a capital do Brasil?"

Deve retornar um objeto

Sentenças que representam objetos são chamados de termos

Fórmulas

São construídos a partir destas regras:

- Todo átomo é uma fórmula da Lógica de Predicados
- Se H é fórmula então (¬H) também é
- Se H e G são fórmulas, então (HvG) também é
- Se H é fórmula e x variável, então

  ((∀x)H) e ((∃x)H) são fórmulas

Correspondência entre quantificadores

Todo piloto é rápido

Equivale

É falso que existe piloto que não é rápido

Existe treinador inteligente

Equivale

É falso que todo treinador não seja inteligente

Correspondência entre quantificadores

$((\forall x)H) = \neg((\exists x)(\neg H))$

$((\exists x)H) = \neg((\forall x)(\neg H))$

Qualquer quantificador pode ser definido a partir do outro!

Lista de exercício

c) As filhas do professor Pedro são lindas e meigas

e) Nem todo pássaro voa

f) todo político é desonesto

n) Quem não se ama não ama ninguém

Lista de exercício

o) Toda patricinha de Petrolina que vai ao shopping tem celular, pele lisa e cheiro de alface

p) Patricinha de Petrolina não gosta de patricinha de Juazeiro

aa) Nenhum filho adolescente de Maria gosta de estudar.

Prolog

•Uma linguagem de **PRO**gramação em **LÓG**ica

•A linguagem Prolog surgiu no início da década de 70

•O Prolog é uma linguagem declarativa que usa um fragmento da lógica de 1ª ordem (as *Cláusulas de Horn*) para representar o conhecimento sobre um dado problema.

Cláusulas de Horn

Cláusulas de Horn são fórmulas na forma

$$p \Leftarrow q1 \wedge q2 \wedge \ldots \wedge qn$$

representadas em Prolog por p :- q1, q2, ..., qn

<cabeça da cláusula> **:-** *<corpo da cláusula>*

Os **fatos** são cláusulas de Horn com o corpo vazio.

O que é um programa em Prolog?

Um programa em Prolog é um “conjunto” de axiomas e de regras de inferência (definindo relações entre objetos) que descrevem um dado problema. A este conjunto chama-se normalmente base de conhecimento.

Como é a execução de um programa em Prolog?

•A execução de um programa em Prolog consiste na **dedução de conseqüências lógicas** da base de conhecimento.

•O usuário faz consultas e o "motor de inferência" do Prolog pesquisa na base de conhecimento por axiomas e regras que permitam (**por dedução lógica**) dar uma resposta.

•O motor de inferência faz a dedução aplicando o algoritmo de **resolução de 1ª ordem**.

Exemplo de um programa Prolog

Factos

Regras

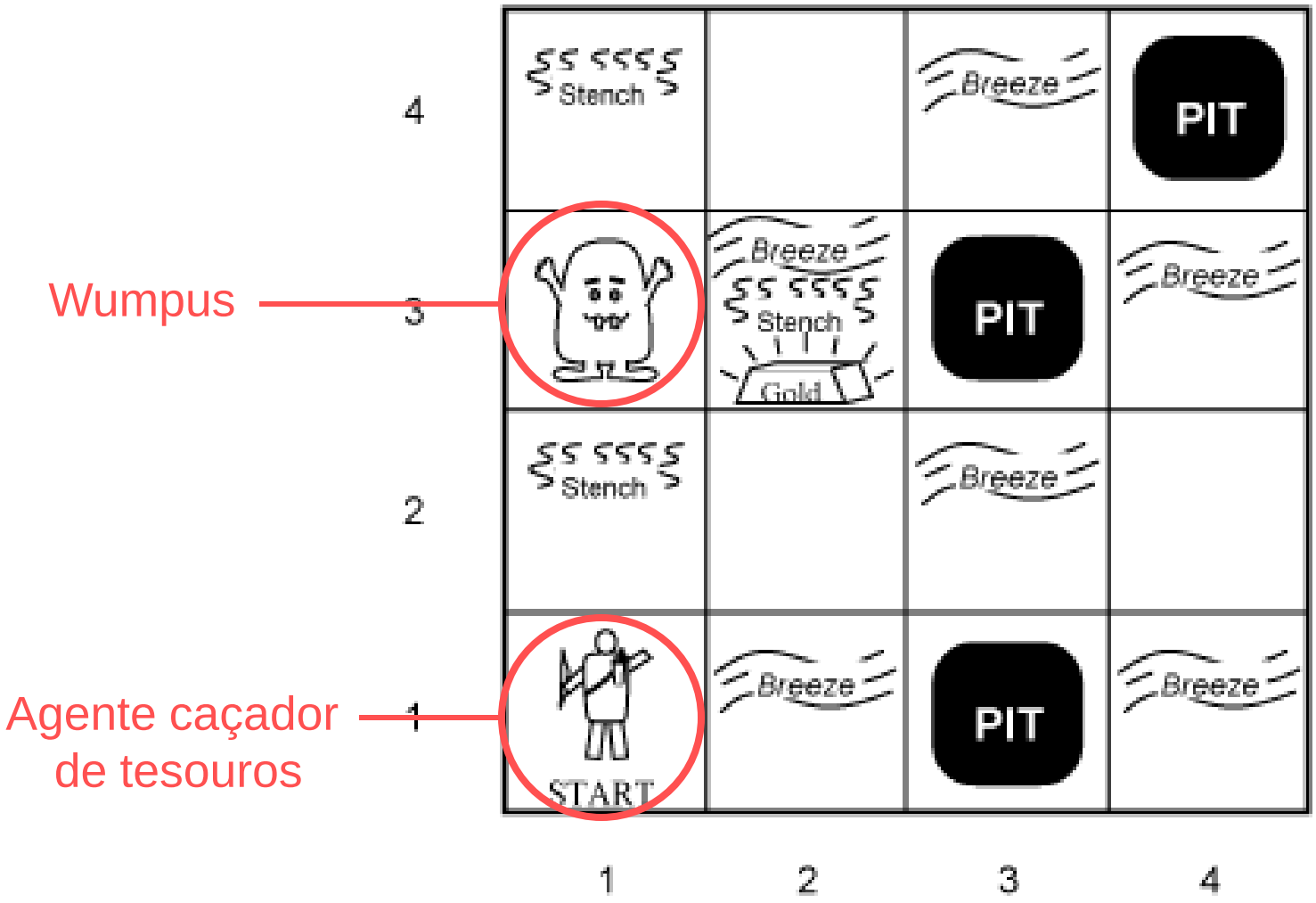
```
mae(sofia,joao).
mae(ana,maria).
mae(carla,sofia).

pai(paulo,luis).
pai(paulo,sofia).
pai(luis,pedro).
pai(luis,maria).

progenitor(A,B) :- pai(A,B).
progenitor(A,B) :- mae(A,B).

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

Mundo Wumpus



Regras em Prolog - Mundo Wumpus

Definir:

- Regra para saber se uma caverna tem buraco
- Regra para saber se uma caverna tem Wumpus
- Regra para saber se uma caverna é segura

Regras em Prolog - Mundo Wumpus

Definir:

- Regra para saber se uma caverna tem buraco

```
% regras para dectar o curaco
buraco(X, Y):-
                XA is X - 1, XP is X + 1, YA is Y - 1, YP is Y + 1,
                (vento(X, YP);choque(X, YP)),
                (vento(X, YA) ; choque(X, YA)),
                (vento(XP, Y) ; choque(XP, Y)),
                (vento(XA, Y); choque(XA, Y)).
```

Regras em Prolog - Mundo Wumpus

Definir:

- Regra para saber se uma caverna tem Wumpus

```
% regras para detectar o wumpus
wumpus(X, Y):-
                XA is X - 1, XP is X + 1, YA is Y - 1, YP is Y + 1,
                (fedor(X, YP);choque(X, YP)),
                (fedor(X, YA) ; choque(X, YA)),
                (fedor(XP, Y) ; choque(XP, Y)),
                (fedor(XA, Y); choque(XA, Y)).
```

Regras em Prolog - Mundo Wumpus

Definir:

- Regra para saber se uma caverna é segura

```
% regra para definir se um lugar é seguro
seguro(X, Y) :- visitado(X, Y).
seguro(X, Y) :- not(buraco(X, Y)), not(wumpus(X, Y)).
```

Aplicação do Mundo Wumpus com Java e Prolog


UNIVASF
FUNDAÇÃO UNIVERSIDADE FEDERAL
DO VALE DO SÃO FRANCISCO