



Inteligência Artificial

PROLOG



Linguagem PROLOG

Enquanto PASCAL é uma linguagem Procedural, baseada em Estruturas lógicas de controle

Prolog é uma linguagem Declarativa, baseada em Lógica matemática

Linguagem PROLOG

LINGUAGENS PROCEDURAIS X LINGUAGENS DECLARATIVAS

PROCEDURAL :

"Tem que dizer ao computador EXATAMENTE COMO FAZER o que se deseja que ele faça"

DECLARATIVA:

"Tem-se que gastar mais tempo dizendo ao computador O QUE você deseja que ele faça e quase nenhum tempo dizendo-lhe O COMO fazer."

"Dá-se ao computador uma DESCRIÇÃO DO PROBLEMA usando em numero de FATOS e REGRAS e então, pergunta-se, para encontrar todas possíveis soluções para o problema."



Linguagem PROLOG

■ Máquina de Inferência do Prolog

O Prolog tem o encadeamento para traz embutido e também utiliza outra técnica, conhecida com BACKTRACKING (rastreamento para trás).

O encadeamento para trás assume uma conclusão (ou consequência) como verdadeira e depois examina um banco de conhecimento de regras/fatos para ver se ele apóia esta suposição.

Se a suposição não estiver correta, o rastreamento para trás é usado para desfazer a suposição e substitui-lá pôr uma nova.

Elementos Básicos

■ Átomo

- é o tipo de *palavra* mais comum em PROLOG
- identifica um *objeto* ou um *predicado*
- começam sempre com letra minúscula
- ex: *ana, caneta, guarda_roupa, josé, livro, manóel, tesoura, robô*
- obs: quando identifica um predicado ou estrutura, é chamado *functor*

■ Predicado

- é o tipo mais comum de *frase afirmativa* em PROLOG
- declara algo a respeito de objetos (propriedade, relação, ...)
- é representado por um *átomo* seguido de objetos entre parêntesis
- ex:

- robô(piter)

← CARACTERISTICA DO OBJETO

- homem(silvestre)

- pai(silvestre,nádia)

← RELAÇÃO ENTRE OS OBJETOS

Variáveis

■ Variáveis Existenciais

- são usadas em frases interrogativas (Quem...?, Qual...?)
- representam objetos desconhecidos, cuja existência se presume
- iniciam sempre com maiúsculas
- ex: ?-robô(*Quem*).

■ Variáveis Universais

- são usadas em frases afirmativas (Qualquer..., Tudo...)
- corresponde ao conceito comum de variável em matemática
- ex: igual(H,H).

■ Variáveis Anônimas

- são usadas em frases compostas, substituindo variáveis universais que não necessitam ser explicitamente referenciadas
- são representadas por um _
- ex: pessoa(X):-pai(X,_);pai(_,X).

Estruturas

■ Estruturas

- são palavras que referenciam objetos compostos por partes individuais
- são definidos no formato *funtor(argumento1,argumento2,...)*
- quanto à codificação, são semelhantes aos predicados
- de fato, todo predicado é uma estrutura que declara um fato (V/F)
- nem toda estrutura é um predicado, algumas só nomeiam objetos, não declarando algo como verdadeiro ou falso

- ex:

data_nascimento(nádia, data(3,março,1986)).

signo(Pessoa,aquário) :- data_nascimento(Pessoa,data(Dia,janeiro,_)),
Dia >= 20.

signo(Pessoa,aquário) :- data_nascimento(Pessoa,data(Dia,fevereiro,_)),
Dia =< 18.

Operadores

■ Declarações *Infixas* (operadores)

- | | | |
|-------|--------------------------------|---------|
| ■ > | maior que | X > 5 |
| ■ < | menor que | X < 5 |
| ■ =< | igual ou menor | X =< 5 |
| ■ >= | maior ou igual | X >= 5 |
| ■ = | igual a | X = 5 |
| ■ \= | diferente | X \= y |
| ■ @< | vem antes de (alfabeto) | a @< b |
| ■ @> | vem depois de (alfabeto) | b @> a |
| ■ @=< | vem antes ou na mesma posição | X @=< Y |
| ■ @>= | vem depois ou na mesma posição | x @>= Y |

■ Operadores lógicos

- not negação
- , *and*, conjunção
- ; *or*, disjunção

Programas

■ Frase

- *cabeça* :- *corpo*.
- *átomo*(arg1, arg2, ...) :- *expressão*.

■ Aridade

- é o número de objetos em um predicado ou de argumentos em uma estrutura
- ex: *data*(13,abril,1960) *aridade* = 3
 pai(edivaldo,daniel) *aridade* = 2

■ Definição

- um conjunto de frases com cabeças contendo o mesmo predicado e mesma aridade

■ Programa

- é o conjunto de várias definições que, juntas, permitem ao computador realizar uma tarefa

Frases

■ Frases Afirmativas Simples

- são constituídas por um único predicado
- terminam com um ponto final `.'
- ex:
 - pai(silvestre,nádia).
 - homem(hadar).
 - ama(hadar,nádia).

■ Frases Interrogativas

- servem para fazer perguntas ao computador
- são codificadas no formato *?-frase*.
- ex:
 - ?-ama(hadar,nádia).
 - ?-robô(piter).
- obs: as respostas obtidas podem ser *yes*, *no*, *->*, *X=xxx*, dependendo da pergunta, e podem admitir interação com ; ou .

Frases

■ Frases Afirmativas Compostas (regras)

- combinam frases simples, geralmente para declarar regras
- formato: *cabeça :- corpo.*
- ex: leis da robótica

proibido_fazer(R,B):-	robô(R), capaz_de_fazer(R,B), machuca(B,H), humano(H).	(1a. Lei)
-----------------------	---	-----------

obrigado_a_fazer(R,B):-	robô(R), capaz_de_fazer(R,B), oferece_perigo(C,para,H), humano(H), afasta(B,C).	(2a. Lei)
-------------------------	---	-----------

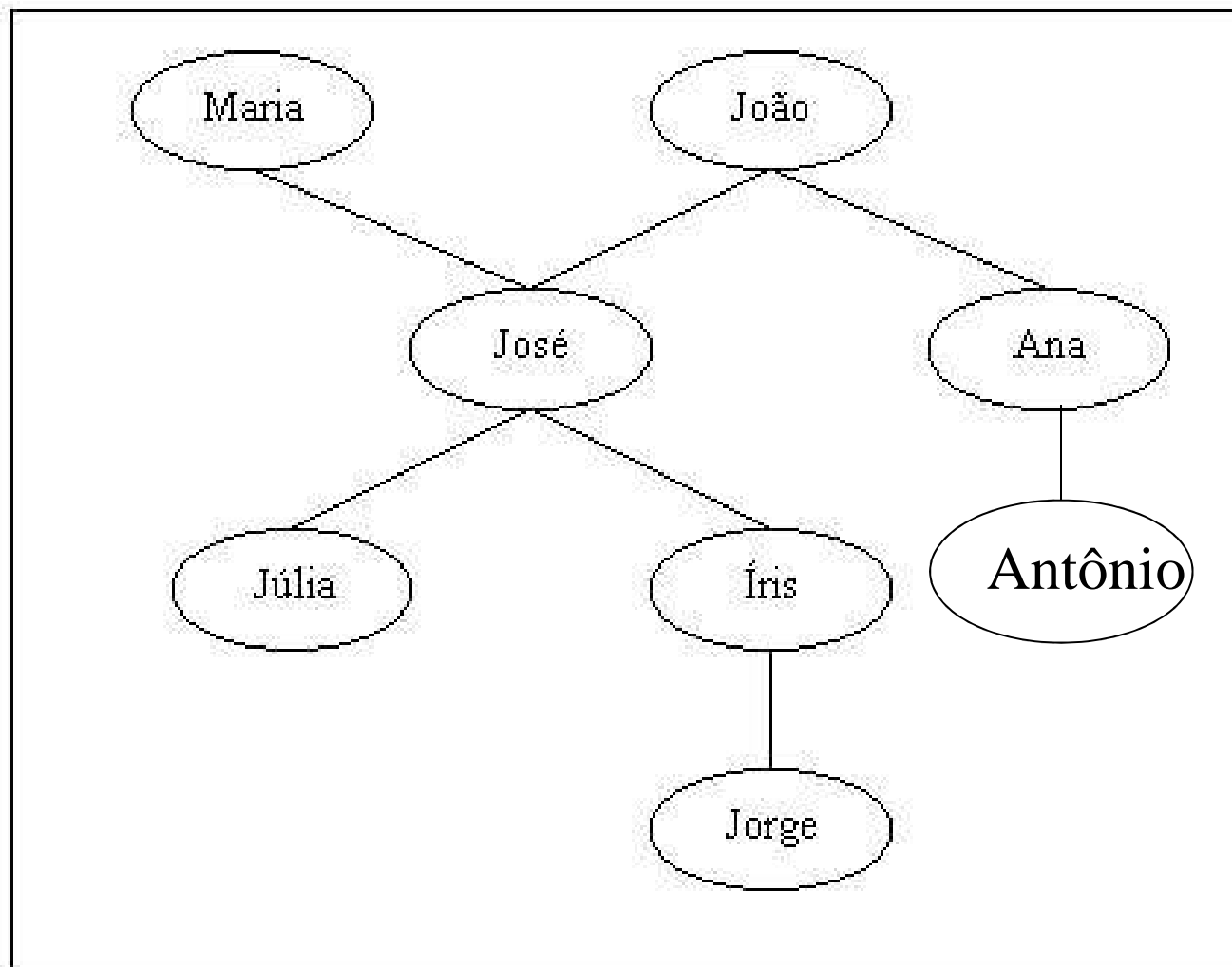
deve_fazer(R,Ato):-	robô(R), capaz_de_fazer(R,Ato), mandou_fazer(H,R,Ato), humano(H), not proibido_fazer(R,Ato).	(3a. Lei)
---------------------	--	-----------

Frases

■ Frases afirmativas usando estruturas

- machuca(derrubar(H),H) :- animal(H).
- animal(H) :- humano(H).
- objeto_físico(O) :- animal(O).
- humano(H):-homem(H);mulher(H).
- oferece_perigo(monstro,para,H) :- humano(H),ataca(monstro,H).
- afasta(derrubar(monstro),monstro).
- mandou_fazer(silvestre,piter,proteger(nádia)).
- mandou_fazer(hadar,piter,derrubar(asrat)).

Exercício



Referências

- Araribóia, G., "Inteligência artificial : um curso prático". Editora(s) LTC, 1989.
- Monard, M. C.; Nicoletti, M. C. "Programas PROLOG para processamento de listas e aplicações". 1993
- Links:
 - <http://kti.ms.mff.cuni.cz/~bartak/prolog/intro.html>