Prolog

Programação Lógica

> Roteiro:

- 1. Surgimento
- 2. Característica
- 3. Tipo de Dados
- 5. Fatos (Base de Dados)
- 6. Consultas
- 7. Regras
- 8. Listas
- 9. Predicados do Prolog
- 10. Exercícios

1. Surgimento

Foi criada em meados de 1972 por Alain Colmerauer e Philippe Roussel, baseados no conceito de Robert Kowalski da interpretação procedimental das cláusulas de Horn. A motivação para isso veio em parte da vontade de reconciliar o uso da lógica como uma linguagem declarativa de representação do conhecimento com a representação procedimental do conhecimento, que era popular na América do Norte no final da década de 1960 para início de 1970.

Aplicações

- Principais aplicações se dão na área de computação simbólica:
-

  " Lógica matemática, prova automática de teoremas e semântica;
  " Solução de equações simbólicas;
  " Bancos de dados relacionais;
  " Linguagem Natural;
  " Sistemas Especialistas;
  " Planejamento Automático de Atividades;
  " Aplicações de "General Problem Solving", como jogos (Xadrez, Damas, Jogo da Velha, etc.);
  " Compiladores;
  " Análise Bioquímica e projetos de novas drogas.

2. Característica

- O Prolog é uma linguagem declarativa, significando que em vez de o programa estipular a maneira de chegar à solução, passo a passo, (como nas linguagens procedimentais ou imperativas), limita-se a fornecer uma descrição do problema que se pretende computar. Usa uma coleção de *fatos* (**base de dados**) e de relações lógicas (***regras***) que exprimem o domínio relacional do problema a resolver.

Introdução ao Prolog

- Características
  - Provador de teoremas (Verdade ou Falso)
  - Linguagem declarativa
  - Linguagem não tipada
  - Linguagem é interpretada
  - Não determinístico
  - Diferente de programação procedimental (definição lógica dos problemas)
  - Não existe variáveis globais
  - Muito usado em IA (rápida prototipação)

Da Notação de Kowalski para um programa Prolog

Notação de Kowalski	Programa Prolog
chama(a,b) ←	chama(a,b).
usa(b,e) ←	usa(b,e).
depende(x,y) ← chama(x,y)	depende(X,Y) :- chama(X,Y).
depende(x,y) ← usa(x,y)	depende(X,Y) :- usa(X,Y).
depende(x,y) ← chama(x,z), depende(z,y)	depende(X,Y) :- chama(X,Z), depende(Z,Y).
← depende(a,e)	?- depende(a,e).

Fato ou Cláusula Unitária

Regra

Questionamento

3.1 Átomos

- As constantes de texto são introduzidas por meio de átomos. Um átomo é uma sequência constituída de letras, números, mas iniciando com uma letra minúscula.

Elementos da Linguagem - Átomos

- Definição: cadeias compostas pelos seguintes caracteres:
  - Letras Maiúsculas: A,..., Z
  - Letras Minúsculas: a,..., z
  - Dígitos: 0, 1,..., 9
  - Caracteres especiais: *, +, _, -, >, <, =, :, , ~
- Composição de Átomos:
  - Cadeias começando com letras minúsculas. Ex.: x_y, maria, curso_de_IA
  - Cadeias de caracteres especiais. Ex.: <---->, ::=
  - Cadeias de caracteres entre apóstrofos. Ex.: ‘Maria’, ‘casa branca’, ‘a’

3.2 Números

- Um número é uma sequência de dígitos, permitindo também os sinais de **.** (para números reais), **-** (número negativo) e **e** (notação científica). Algumas implementações do Prolog não fazem distinção entre inteiros e números reais.

Ex: 321 3.21

3.3 Variáveis

- Variáveis são declaradas da mesma forma que átomos, porém iniciando com uma letra maiúscula. No ambiente Prolog uma variável não é um contêiner cujo valor pode ser atribuído (como ocorre nas linguagens imperativas). Seu comportamento é mais próximo de um padrão, que é incrementalmente especificado pela unificação. Em outras palavras, uma variável Prolog é como uma incógnita, cujo valor é desconhecido a princípio, mas após descoberto, não sofre mais mudanças.

Ex: Ana X _X _Ana _ana

Programa Prolog

- Declaração de *fatos* (cláusulas unitárias)
- Declaração de *regras*
- Interrogação a respeito desses elementos

Fatos e regras denotam relações entre objetos

Programa Prolog (cont.)

- Estrutura de um fato

- Estrutura de uma regra

```
gosta(joao, X) :-
        gosta(X, vinho),
        gosta(X, futebol).
```

Conjunção

Questionamentos

- Dada a **base de fatos**:

```
gosta(julio,peixe).
gosta(julio,maria).
gosta(maria,livro).
gosta(joao,livro).
```

Quem gosta de livro?
?- gosta(X, livro).

Quem gosta de livro e chocolate?
?- gosta(X, livro), gosta(X, chocolate).

Quem gosta de livro ou chocolate?
?- gosta(X, livro); gosta(X, chocolate).

Disjunção

Introdução ao SWI-Prolog (cont.)

- Operadores relacionais
  - X = Y — X e Y são iguais;
  - X \= Y — X e Y são diferentes;
  - X < Y — X é menor que Y;
  - X > Y — X é maior que Y;
  - X =< Y — X é menor ou igual a Y;
  - X >= Y — X é maior ou igual a Y.
  - X =:= Y — X e Y são iguais (p/ números);
  - X =\= Y — X e Y são diferentes (p/ números).

Introdução ao SWI-Prolog (cont.)

- Operadores aritméticos
  - X+Y soma de X e Y;
  - X – Y diferença de X e Y;
  - X * Y multiplicação de X por Y;
  - X / Y divisão de X por Y;
  - X mod Y resto da divisão de X por Y.
- Atribuição de valores numéricos “is”:

  ?- X is 10 + 2.
- Negação de predicados:

  \+ não

4. Operadores

Operador	Símbolo	Exemplo
E	,	A , B
OU	;	A ; B
Negação	\+	\+ A
Igualdade	=	A = B
Diferença	\==	A \== B

5. Fatos (Base de Dados)

- A unidade básica do Prolog é o predicado, que é postulado verdadeiro. Um predicado consiste de uma cabeça e um número de argumentos.

```
1
2   %/ Base de Dados%/
3
4   animal(cachorro).%Animal é a cabeça, e cachorro o predicado
5   pai(jose, antonio).%pai é a cabeça, e jose e antonio são os predicados
6
```

Linha 5 se lê: **cachorro** é um **animal**

Linha 6 se lê: **jose** é **pai** de **antonio**

6. Consultas

- Para recuperar informações de um programa lógico, usamos consultas. Uma consulta pergunta se uma determinado relacionamento existe entre objetos.

```
19 ?- animal(cachorro).
true.

20 ?- animal(gato).
false.

21 ?- animal(X).
X = cachorro.
```

Não Está na Base de Dados

Linha 19: cachorro é um animal?
Linha 20: gato é um animal?
Linha 21: Quem é animal?

7. Regras

- O segundo tipo de predicado no Prolog é a regra, também chamada de "cláusula".
- Ex:

```
luz(acesa) :- interruptor(ligado).
```

A luz está acesa se o interruptor estiver ligado.

Obs: ‘ :- ’ significa ‘ SE ’

NomeRegra(Varivável(is)) :- Condições .

- Regras:
  - Para utilizarmos uma regra , se usa o símbolo “:-”

Ex.:

Dados os fatos:

pai(arthur,silvio).
pai(arthur,carlos).
pai(carlos,xico).
pai(silvio,ricardo).

Utilizaremos a seguinte regra:

avo(X,Z) :- pai(X,Y), pai(Y,Z).

Isso significa que se alguém é pai de uma pessoa, que por sua vez é pai de outra pessoa, então ele é avô.

Vamos realizar uma querie para conferir a regra:

?- avo(arthur,xico),avo(arthur,ricardo).

Resposta : “YES”

Prática – Criar a base de dados

pais e mães

```
pai(antonio, rui).          (o antonio é pai do rui)
pai(antonio, sandra).
mae(maria, rui).            (a maria é mãe do rui)
mae(maria, sandra).

feminino(maria).
feminino(sandra).
masculino(antonio).
masculino(rui).
```

Sintaxe

Quando o corpo não é vazio as cláusulas designam-se por regras, definindo relações à custa de outras relações.

filhos e filhas

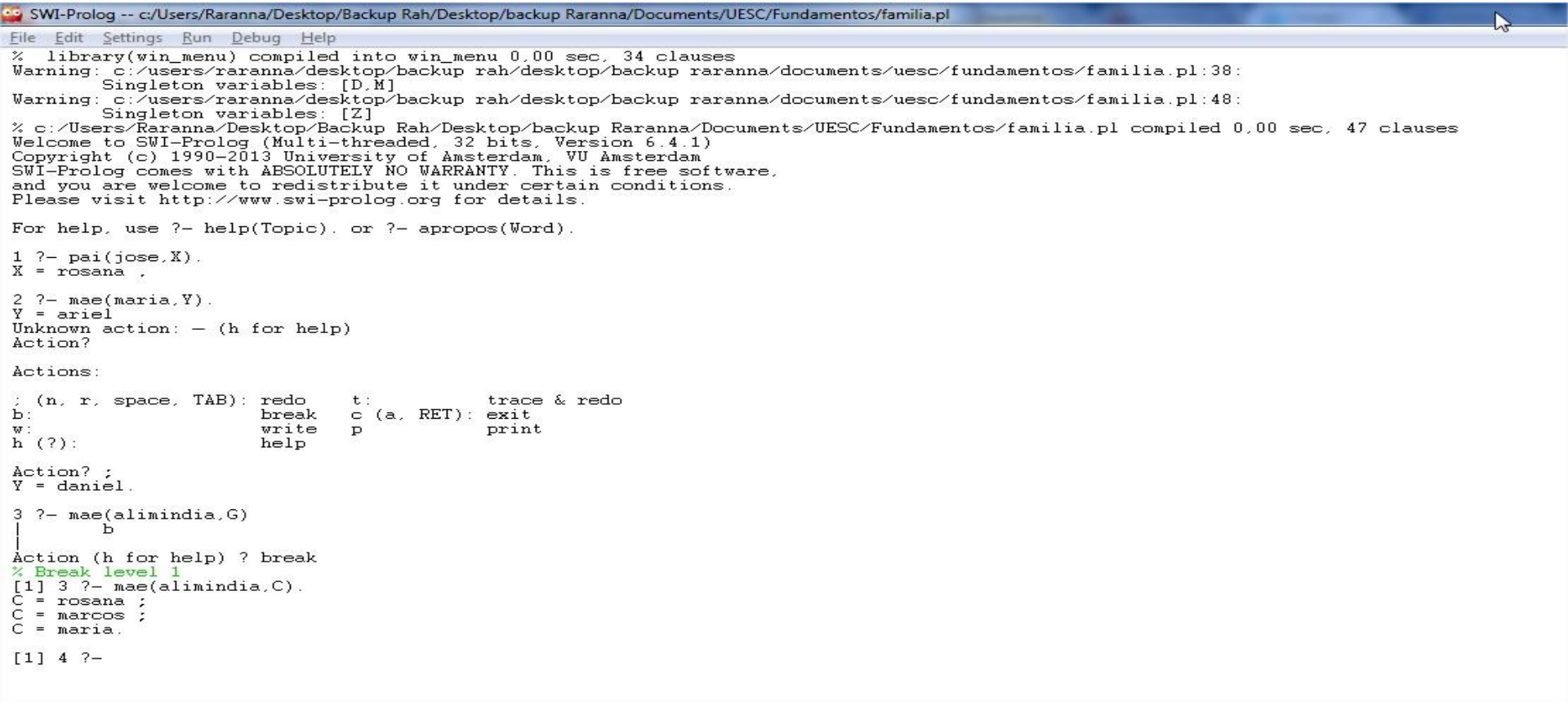
```
filho(X,Y) :- pai(Y,X), masculino(X).
filho(X,Y) :- mae(Y,X), masculino(X).

filha(X,Y) :- pai(Y,X), feminino(X).
filha(X,Y) :- mae(Y,X), feminino(X).
```

ou

Executando um programa

- Software Swi-Prolog 6.4.1 plataforma Windows :

  - Criação de um novo arquivo fonte .pl
  - Edição de .pl já existente
  - Execução de arquivo finalizado



Como podemos ver, o Prolog é uma linguagem muito poderosa, principalmente na área de Inteligência Artificial onde é líder absoluta. Entre as implementações do Prolog, podemos citar o Visual Prolog (Turbo Prolog), o SWI Prolog, GNU Prolog, Amzi! Prolog, entre muitas outras já existentes.