Universidade Federal da Bahia
Instituto de Computação – DCC/UFBA

Aula 05 – PROLOG – Introdução e Elementos Básicos

Disciplina: MATA56 - Paradigmas de Linguagens de Programação

Profº Claudio Junior Nascimento da Silva

claudiojns@ufba.br

Agenda

- Histórico
- Elementos básicos:
  - Fatos
  - Regras
  - Cláusula
  - Predicado
  - Consultas
  - Sintaxe
  - Variáveis
- Exercícios

Programação Lógica

- Ideia de programas que pudessem manipular sentenças baseadas em uma linguagem formal, como cálculo de predicados (MacCarthy, 1958);
- Expressar programas na forma de lógica simbólica e usar um processo de inferência lógica para produzir resultados;
- Programador
  - Declara teoremas ou objetivos;
  - Implementa linguagem;
  - Busca encontrar fatos e passos de inferência;
  - Fatos e inferência implicam no objetivo.
- Aplicações: IA, PLN, Sistemas especialistas e prova de teoremas;
- Linguagem mais popular: Prolog

Histórico

- Prolog: Programming in Logic;
- Baseado em Lógica de Predicados;
- O fundamento básico por trás do Prolog é a noção de programação em lógica, onde o processo de computação pode ser visto como uma sequência lógica de inferências;
- Programa Prolog:
  - Coleção de fatos e de regras que definem relações entre os objetos do discurso de um problema.
  - Envolve a dedução de consequências a partir de regras e fatos.
- Programação Lógica (PL) não é sinônimo de Prolog;
- PL possui duas grandes classes de programas:
  - Cláusulas definidas;
  - Cláusulas não-definidas.
- Prolog é um sistema que executa programas para cláusulas definidas.

Histórico – Linha do tempo

- 1965 – Alan Robinson desenvolveu componentes-chave de um provador de teoremas para lógica de cláusulas (lógica de primeira ordem): procedimento de resolução e o algoritmo de unificação;
- 1972 – R. Kowalski formulou a interpretação procedimental para as cláusulas de Horn: uma clausula lógica, que equivale a uma implicação lógica, pode ser lido e executado como um procedimento em uma linguagem de programação recursiva:

prog if ler(Dado) and calcular(Dado, Result) and impr(Result)

```
procedure prog;
  begin
    ler(Dado);
    calcular(Dado, Result);
    impr(Result);
  end
```

- 1973 – Alain Colmerauer, Universidade de Aix-Marseille desenvolveu um provador de teoremas para implementar sistemas de Processamento de Linguagem Natural (PLN), chamado Prolog (Programation et Logique)

Elementos básicos - Sintaxe

Tipo	Descrição	Exemplo
variáveis	Iniciadas com letras maiúsculas ou (_), seguida de qualquer caractere alfanumérico	X, Y1, _Nome
variável anônima	Definida apenas com (_). Não se deseja saber seu valor	_
átomos	São constantes expressas por palavras. Devem ser iniciadas com letras minúsculas, seguidas de qualquer caractere alfanumérico. Caso seja necessário definir um átomo com letra maiúscula ou número, deve-se usar aspas simples	Joao, 'João', '16'
inteiros	Qualquer número que não contenha um ponto (.). Caractere ASCII entre "" (aspas duplas) são considerados inteiros	1, 5, -3, "a"
floats	Qualquer número com um ponto e pelo menos uma casa decimal	5.3, 7.8, 7.
listas	Sequência ordenada de elementos entre[] e separadas por vírgulas	[a,b,c], [a | b,c]

Elementos básicos - Sintaxe

Comandos	Descrição	Exemplo
write	Para escrever, basta utilizar o comando write(+termo)	write('Teste de impressão.'). write(Teste de impressão.). write(X). write(joao).
read	Para ler, basta utilizar o comando read(+var)	read(X). read(x). read(Joao). read(joao).
atom	Verifica se é um átomo	atom(joao).
var	Verifica se é uma variável	var(Joao)
number	Verifica se é um número	number(12)
is_list	Verifica se é uma lista	is_list([])

Elementos básicos - Sintaxe

Caracteres especiais	Descrição
nl, \n, \l	Nova linha
\r	Retorna ao início da linha
\t	Tabulação
\%	Imprime o símbolo %
%	Todo o texto existente na mesma linha após o símbolo % é considerado comentário
/* */	Todo o texto entre os símbolos é considerado comentário

Programa Prolog

- Conjunto de todas as consequências deduzíveis pela iterativa aplicação das regras sobre os fatos iniciais e os novos fatos gerados.

Programa Prolog é uma coleção de unidades lógicas chamadas de predicados → Cada **predicado** é uma coleção de cláusulas → Uma cláusula é uma regra ou um fato

Declarar fatos a respeito de objetos e seus relacionamentos → Definir regas sobre os objetos e seus relacionamentos → Fazer perguntas sobre os objetos e seus relacionamentos

Fixando conceitos básicos

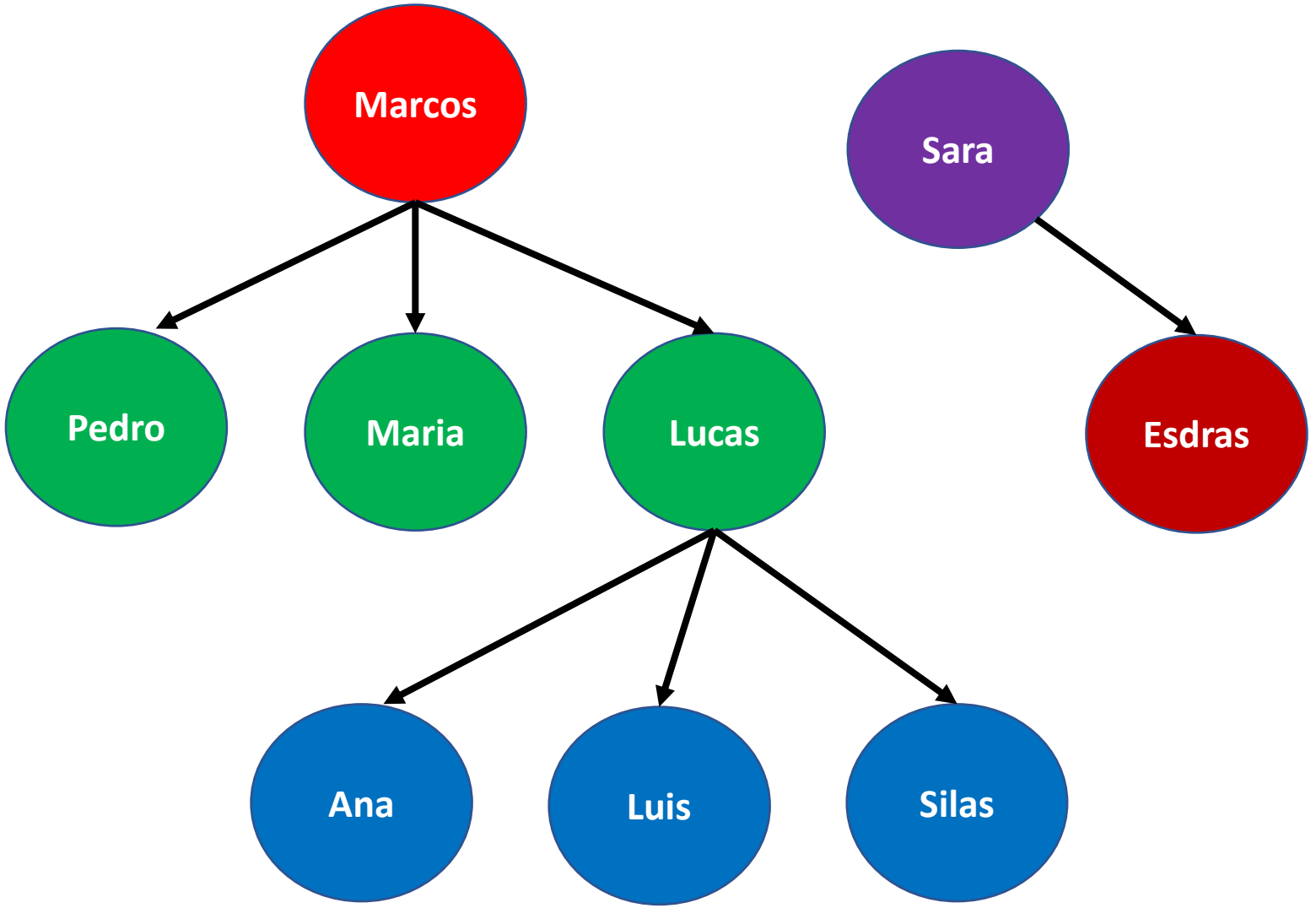
Prolog - Prática

- Site: https://www.swi-prolog.org/
- Instalação (Linux): sudo apt-get install swi-prolog
- Interpretador online: http://swish.swi-prolog.org/

Elementos básicos

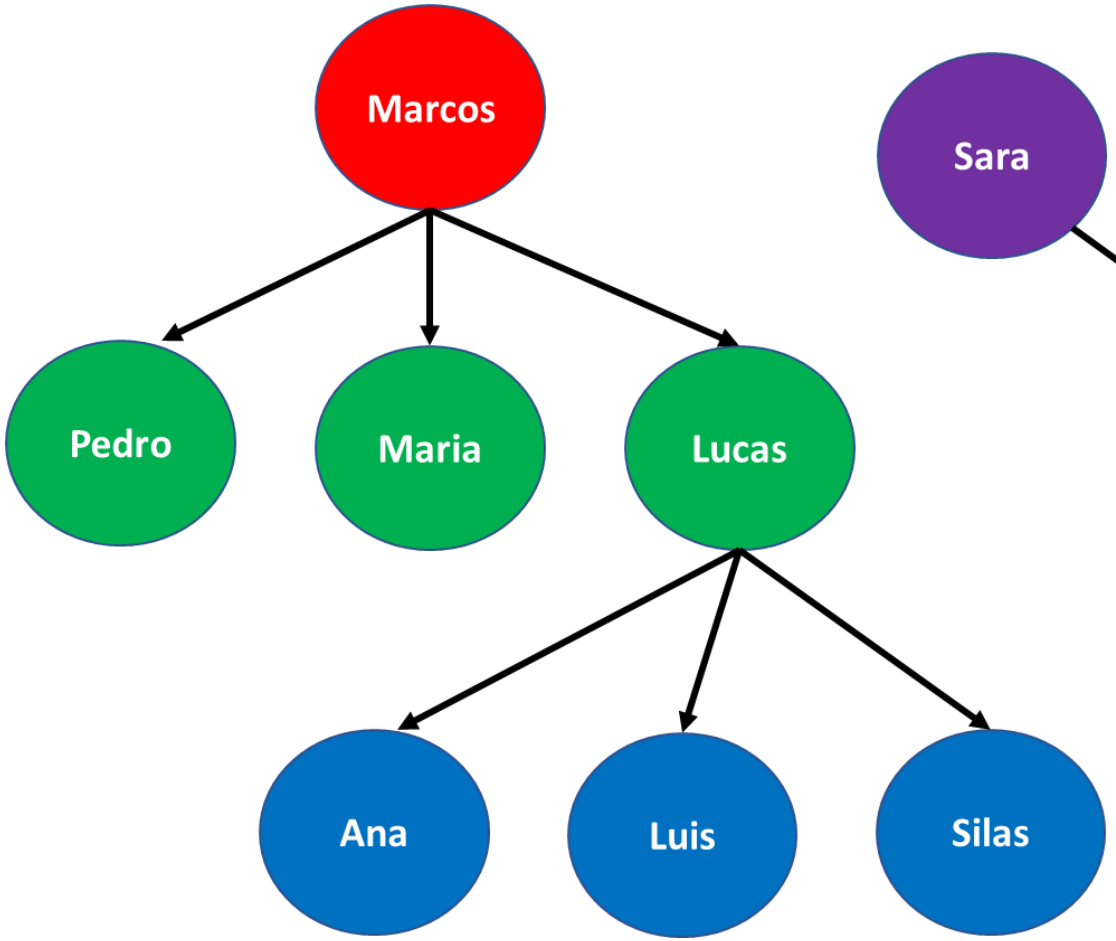
- Fatos – Estabelecem um relacionamento entre objetos de um determinado contexto de discurso:
  - `pai(marcos, pedro).`
- Regras – Permitem definir novas relações em termos de outras relações já existentes:
  - `irmao(X,Y):-pai(P,X),pai(P,Y), X\==Y.`
- Fatos e Regras são tipos de cláusulas.
- Consultas – Usadas para recuperar informações de um programa. Pergunta se um determinado relacionamento existe entre os objetos:
  - `?- fem(sara).`

Exemplo dirigido 01



Exemplo dirigido 01

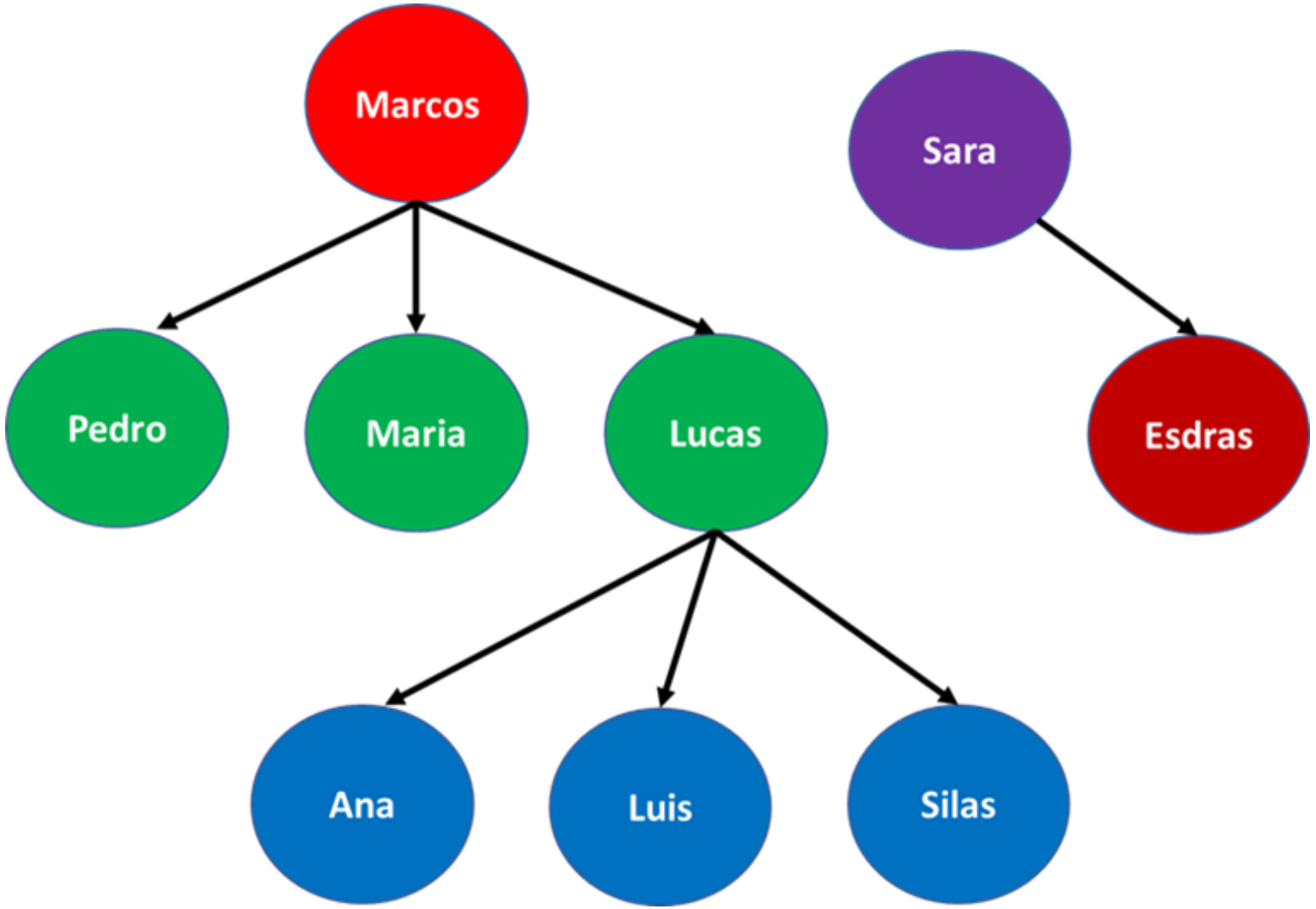
- Quais o fatos podem ser obtidos a partir do Exemplo 01



- Marcos é pai de Pedro ←→ pai(marcos,pedro).
- Marcos é pai de Maria ←→ pai(marcos,maria).
- Marcos é pai de Lucas ←→ pai(marcos,lucas).
- Lucas é pai de Ana ←→ pai(lucas,ana).
- Lucas é pai de Luis ←→ pai(lucas,luis).
- Lucas é pai de Silas ←→ pai(lucas,silas).
- Sara é mãe de Esdras ←→ mae(sara,esdras).

Exemplo dirigido 01

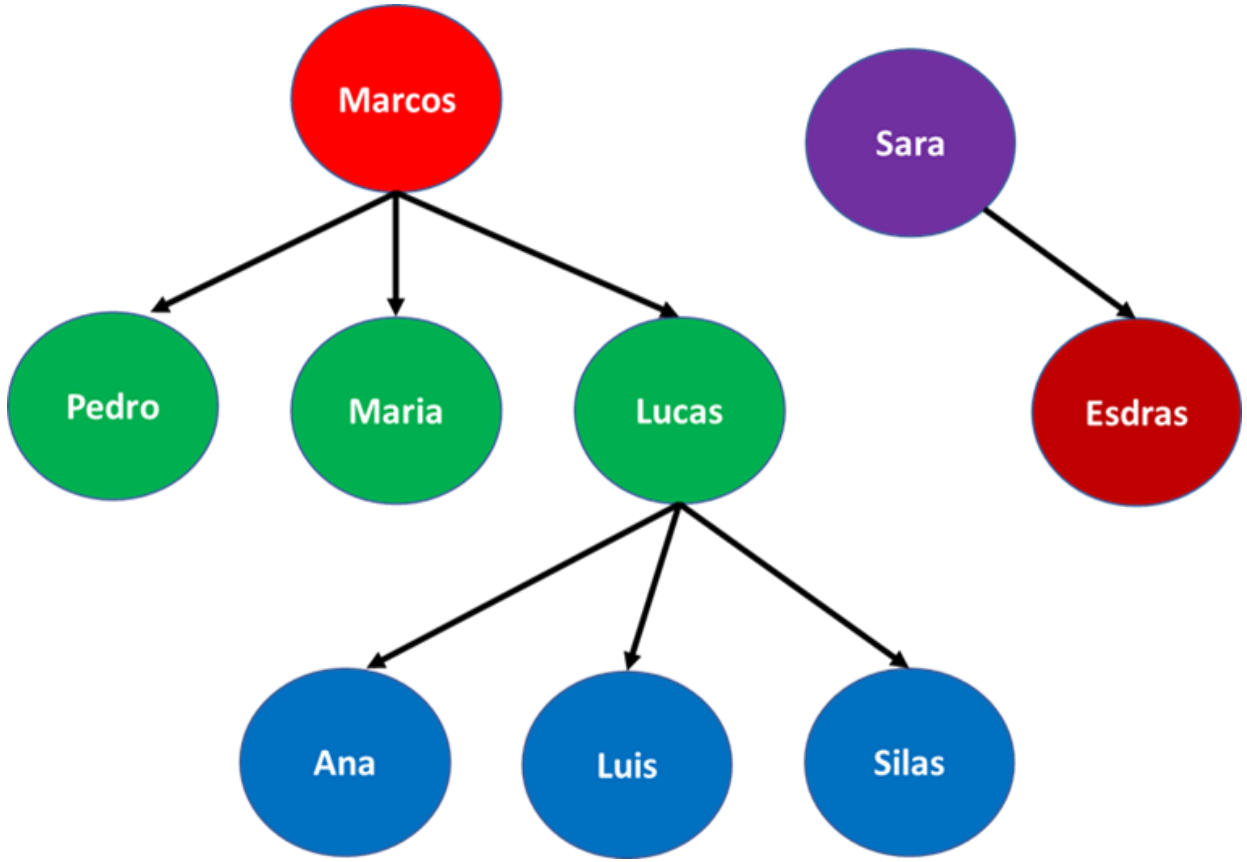
- Quais o perguntas/consultas podemos fazer a partir do Exemplo 01



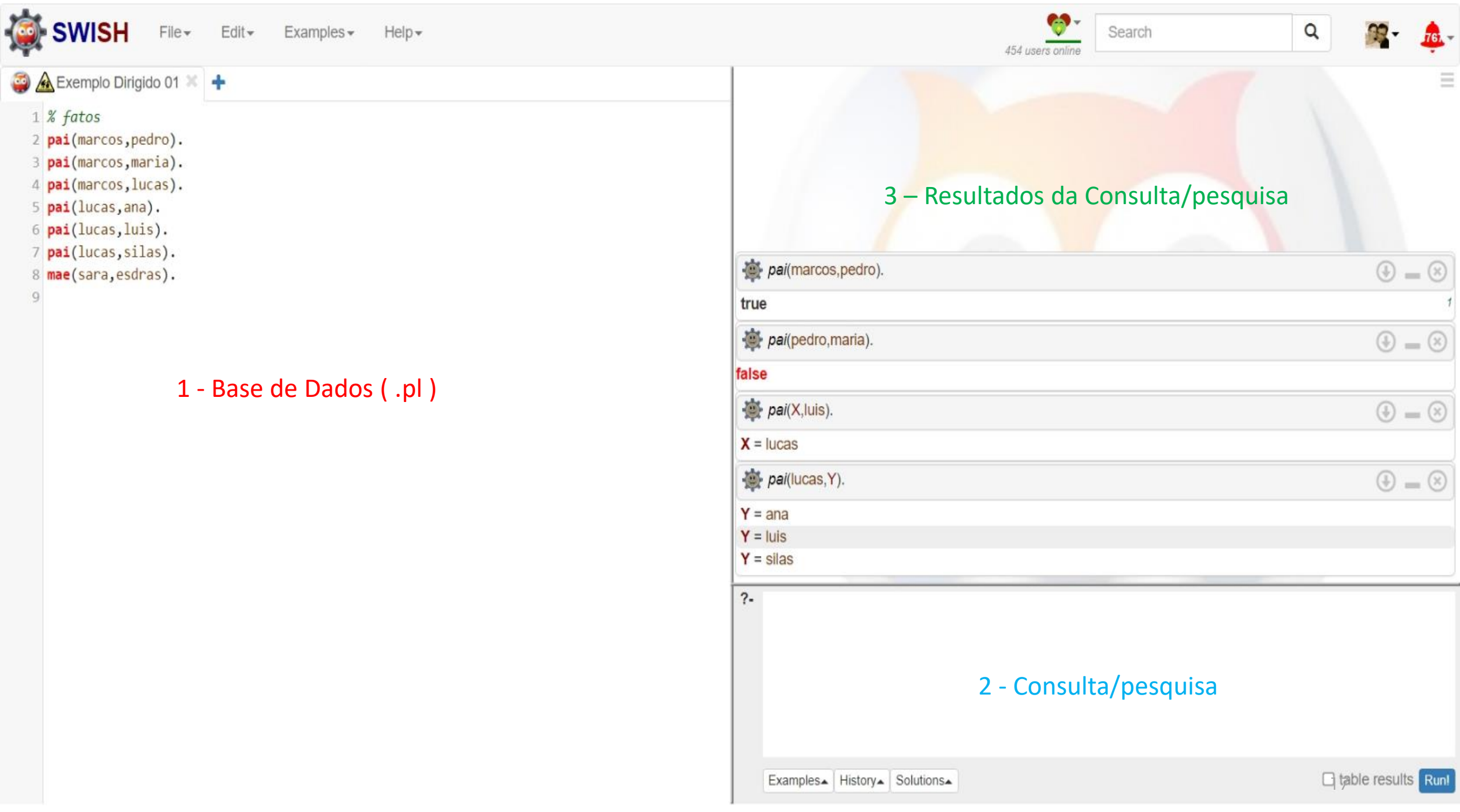
- Marcos é pai de Pedro?
- Pedro é pai de Maria?
- Quem é o pai de Luis?
- Quem são os filhos de Lucas?
- Quem são os irmãos de Maria?
- Ana é irmã de Luís?
- Quem é o avô de Silas?
- Sara é do sexo feminino?
- ~~Quem é a mãe de Maria?~~
- Quem é a mãe de Esdras?
- ~~Ana é do sexo feminino?~~
- Maria é prima de Silas?

Exemplo dirigido 01

- Quais o perguntas/consultas podemos fazer a partir do Exemplo 01

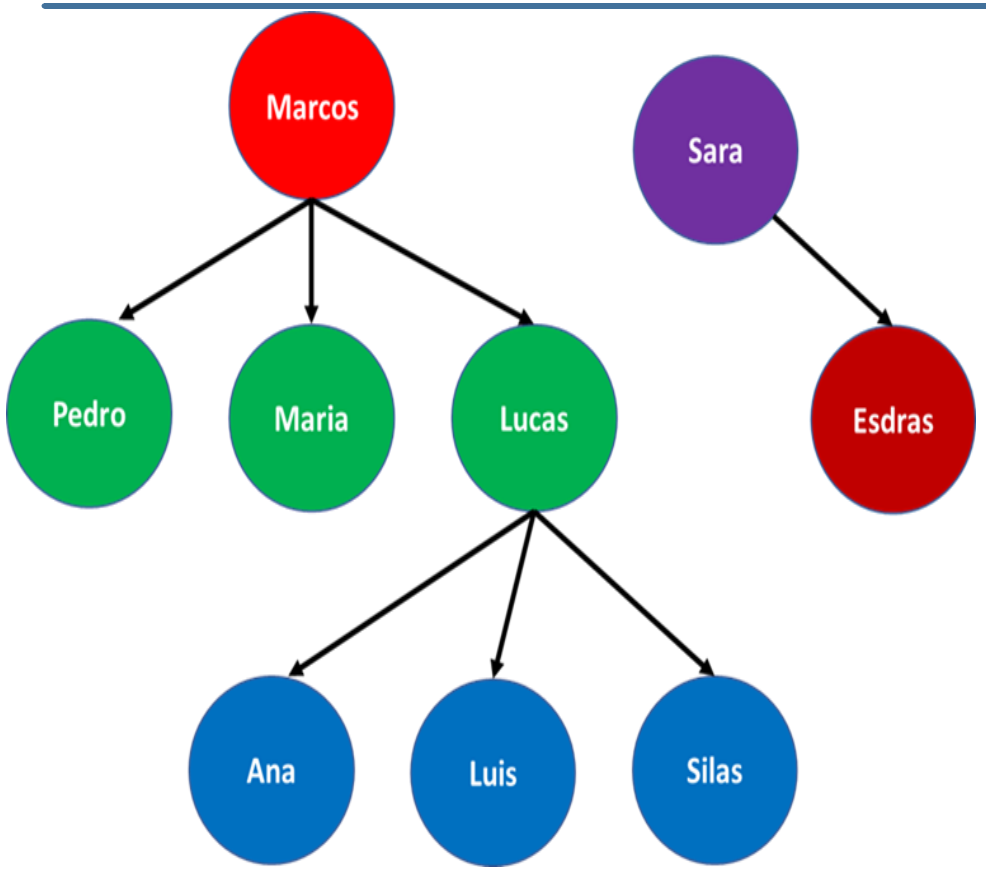


Perguntas	Consultas
Marcos é pai de Pedro?	*pai(marcos,edro).*
Pedro é pai de Maria?	*pai(pedro,maria).*
Quem é o pai de Luis?	*pai(X,luis).*
Quem são os filhos de Lucas?	*pai(lucas,Y).*
Quem são os irmãos de Maria?	
Ana é irmã de Luís?	
Quem é o avô de Silas?	
Sara é do sexo feminino?	
~~Quem é a mãe de Maria?~~	
Quem é a mãe de Esdras?	
~~Ana é do sexo feminino?~~	
Maria é prima de Silas?	


SWISH
File
Edit
Examples
Help
454 users online
Search
Exemplo Dirigido 01
% fatos
pai(marcos,pedro).
pai(marcos,maria).
pai(marcos,lucas).
pai(lucas,ana).
pai(lucas,luis).
pai(lucas,silas).
mae(sara,esdras).
1 - Base de Dados (.pl)
3 – Resultados da Consulta/pesquisa
pai(marcos,pedro).
true
pai(pedro,maria).
false
pai(X,luis).
X = lucas
pai(lucas,Y).
Y = ana
Y = luis
Y = silas
?-
2 - Consulta/pesquisa
Examples
History
Solutions
table results
Run!

Exemplo dirigido 01

- Quais o perguntas/consultas podemos fazer a partir do Exemplo 01



Perguntas	Consultas
Marcos é pai de Pedro?	*pai(marcos,pedro).*
Pedro é pai de Maria?	*pai(pedro,maria).*
Quem é o pai de Luis?	*pai(X,luis).*
Quem são os filhos de Lucas?	*pai(lucas,Y).*
Quem são os irmãos de Maria?	irmaos(X,Y) :- pai(Z,X) , pai(Z,Y) , X \== Y. *irmaos(maria,Y).*
Ana é irmã de Luís?	irmaos(ana,luis).
Quem é o avô de Silas?	avo(X,Y) :- pai(Z,X) , pai(Y,Z). avo(silas,Y).
Sara é do sexo feminino?	fem(X) :- mae(X,_). fem(sara).
~~Quem é a mãe de Maria?~~	??????????
Quem é a mãe de Esdras?	mae(X,esdras).
~~Ana é do sexo feminino?~~	??????????
Maria é tia de Silas?	tio(X,Y) :- pai(Z,Y) , pai(W,Z), pai(W,X), Z \== X. tio(maria,silas).

SWISH File Edit Examples Help

530 users online

Search

Exemplo Dirigido 01

```
1  % fatos
2  pai(marcos,pedro).
3  pai(marcos,maria).
4  pai(marcos,lucas).
5  pai(lucas,ana).
6  pai(lucas,luis).
7  pai(lucas,silas).
8  mae(sara,esdras).
9
10 %regras
11 irmaos(X,Y) :- pai(Z,X) , pai(Z,Y) , X \== Y.
12 avo(X,Y)    :- pai(Z,X) , pai(Y,Z).
13 fem(X)      :- mae(X,_).
14 tio(X,Y)    :- pai(Z,Y) , pai(W,Z), pai(W,X), Z \== X.
15
16
17
18
19
20
21
```

1 - Base de Dados (.pl)

Y = ana
Y = luis
Y = silas

irmaos(maria,Y).

Y = pedro
Y = lucas

3 – Resultados da Consulta/pesquisa

irmaos(ana,luis).

true

avo(silas,Y).

Y = marcos

fem(sara).

true

mae(X,esdras).

X = sara

tio(maria,silas).

true

tio(X,silas).

X = pedro
X = maria

?- tio(X,silas).

2 - Consulta/pesquisa

Examples History Solutions table results Run!

SWI-PROLOG

Windows

SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)

File Edit Settings Run Debug Help

```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- tio(X,silas).
X = pedro ;
X = maria ;
false.

?- pai(lucas,Y).
Y = ana ;
Y = luis ;
Y = silas.
```

2 - Consulta/pesquisa

3 – Resultados da Consulta/pesquisa

2 - Consulta/pesquisa

3 – Resultados da Consulta/pesquisa

exemplo dirigido 01.pl [modified]

File Edit Browse Compile Prolog Pce Help

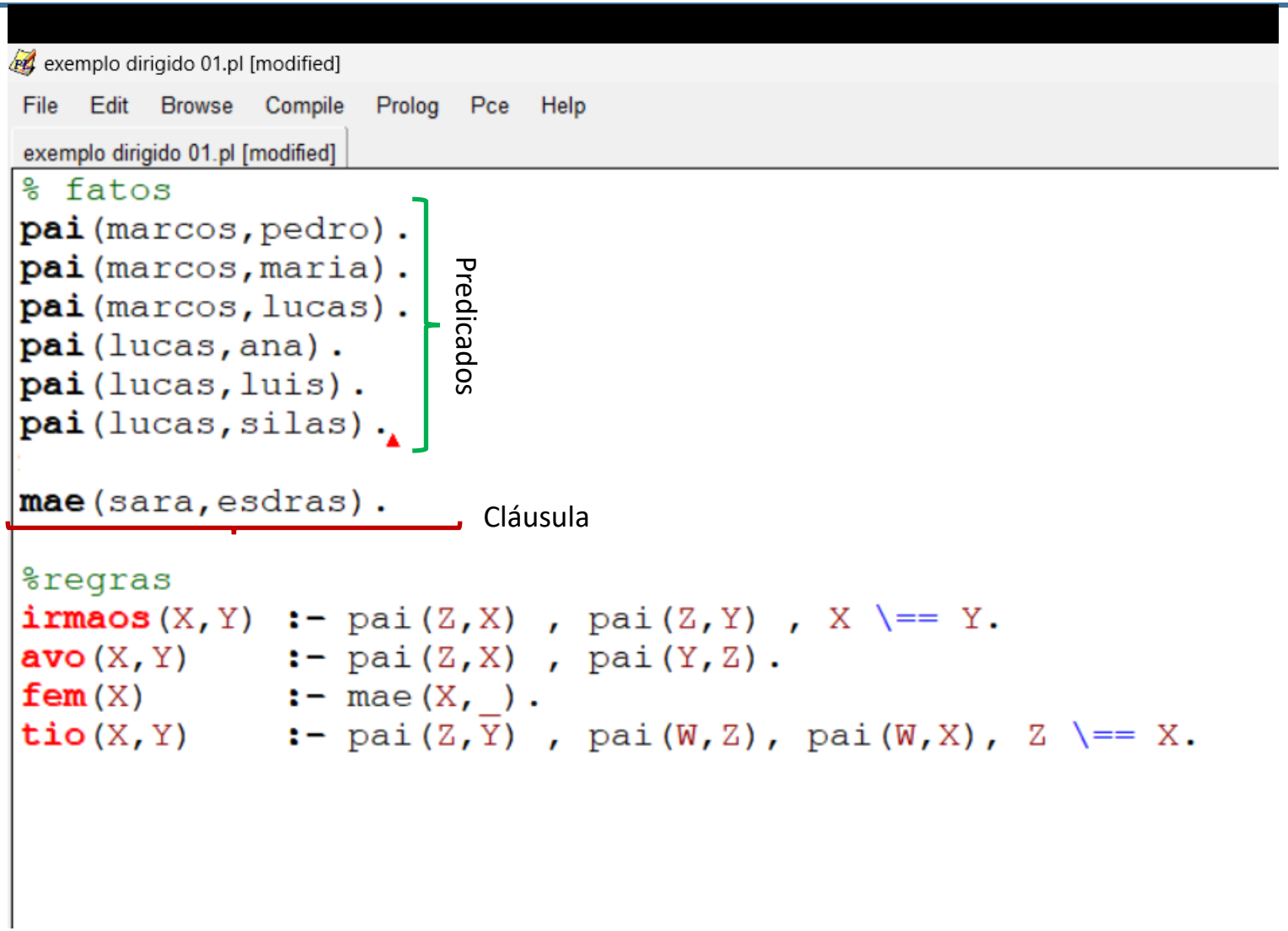
exemplo dirigido 01.pl [modified]

```
% fatos
pai(marcos,pedro).
pai(marcos,maria).
pai(marcos,lucas).
pai(lucas,ana).
pai(lucas,luis).
pai(lucas,silas).
mae(sara,esdras).

%regras
irmaos(X,Y)  :- pai(Z,X) , pai(Z,Y) , X \== Y.
avo(X,Y)     :- pai(Z,X) , pai(Y,Z).
fem(X)       :- mae(X,_).
tio(X,Y)     :- pai(Z,Y) , pai(W,Z), pai(W,X), Z \== X.
```

1 - Base de Dados (.pl)

Estrutura Prolog



Perguntas:

exemplo dirigido 01.pl [modified]

File Edit Browse Compile Prolog Pce Help

exemplo dirigido 01.pl [modified]

```
% fatos
pai(marcos,pedro).
pai(marcos,maria).
pai(marcos,lucas).
pai(lucas,ana).
pai(lucas,luis).
pai(lucas,silas).
mae(sara,esdras).

%regras
irmaos(X,Y)  :- pai(Z,X) , pai(Z,Y) , X \== Y.
avo(X,Y)     :- pai(Z,X) , pai(Y,Z).
fem(X)       :- mae(X,_).
tio(X,Y)     :- pai(Z,Y) , pai(W,Z), pai(W,X), Z \== X.
```

1. Como fazer para determinar o sexo de uma pessoa?
2. Existe outra forma de escrever a regra tio? Em caso de positivo, qual?
3. Como escreveríamos uma regra para consultar/pesquisar primos?
4. Podemos ter fatos e regras com o mesmo nome? Prove.

Fatos, Regras e Consultas

Fatos

- Os fatos são afirmações consideradas verdadeiras.
- Estabelecem um relacionamento entre objetos de um determinado contexto de discurso:
  - homem(x). - significa que "x é um homem";
  - genitor(x, y). – significa que "x é genitor de y" ou "y é gerado de x";
  - timedefutebol(x). – significa que "x é um time de futebol";
  - timeestado(x,y). – significa que "x é um time do estado y";
  - aluno(joao). – significa que "joao é um aluno";
  - professor(Pedro). – significa que "pedro é um professor";
  - curso(computacao). – significa que "computação é um curso";
  - estudadisciplia(joao, computacao). – significa que "joao estuda a disciplina computação";
  - lecionadisciplina(pedro,logica). – significa que "pedro é professor de lógica";

Consultas

- Para responder consultas o Prolog utiliza:
  - matching : checa se determinado padrão está presente, para saber quais fatos e regras podem ser utilizados;
  - unificação : substitui o valor das variáveis para determinar se a consulta é satisfeita pelo fatos ou regras da base (programa);
  - resolução : verifica se uma consulta é consequência lógica dos fatos e regras da base (programa);
  - recursão : utiliza regras que chamam a si mesmas para realizar demonstrações;
  - backtracking : para checar todas as possibilidades de resposta.

Regras

- Os fatos são sempre verdadeiros, mas as regras precisam ser avaliadas. A partir das regras se deduzem fatos não-declarados;
- As regras facilitam a execução de consultas e tornam um programa mais expressivo;
- Uma cláusula Prolog é equivalente a uma fórmula lógica de 1ª ordem. Em Prolog existem os seguintes conectivos:

Conectivo	Prolog	Descrição
se	:-	Equivalente à implicação
e	,	Equivalente à conjunção
ou	;	Equivalente à disjunção

- A fórmula $A(x) \rightarrow B(x) \vee (C(x) \wedge D(x))$ em Prolog: a(X) :- b(X); (c(x) , d(x)).

Regras

- Estrutura da regra:
  - corpo(+arg) :- condicao1(+arg) {[,] ou [;]} condicao2(arg)... [.]
- Regra 01:
  - Regra: `filho(X,Y) :- genitor(Y,X).`
  - Consulta: `?- filho(pam, bob).`
- Regra 02:
  - Regra: `irmao(X,Y):- pai(P,X) , pai(P,Y) , X \== Y .`
  - Consulta: `?- irmao(melca, lot).`

Consultas

```
homem(tom).
mulher(pam).
genitor(pam, bob).
genitor(tom, bob).
```

```
?- genitor(tom, X). % tom é genitor de quem (X)?
X = bob;        <- unifica(X/bob)
Yes             <- genitor(tom, bob). (matching)
No              <- busca outras soluções (backtrack)
                <- nenhum outro fato satisfaz a consulta
```

```
?- genitor(X, bob). % Quem (X) é/são o(s) genitores de bob?
X = tom;        <- unifica(X/tom)
Yes             <- genitor(tom, bob). (matching)
                <- busca outras soluções (backtrack)
X = pam;        <- unifica(X/pam)
Yes             <- genitor(pam, bob).  (matching)
No              <- busca outras soluções (backtrack)
                <- nenhum outro fato satisfaz a consulta
```