



Universidade Federal de Santa Catarina
Campus Araranguá
Engenharia de Computação
ARA7502 – Lógica Aplicada a Computação
Prof. Gustavo Mello Machado

Trabalho Prolog 01 - Aula 27/10/2016

Orientações preliminares.

- É permitida a realização deste trabalho individualmente ou em duplas.
- As entregas serão aceitas exclusivamente via Moodle
- Este trabalho comporá a nota da avaliação E1 como previsto no plano de ensino.

AMBIENTE WEB PARA SWI-PROLOG

Para seguir o tutorial e realizar os exercícios propostos neste trabalho acesse a plataforma WEB de desenvolvimento Prolog disponível no seguinte link:

<http://swish.swi-prolog.org/>

FATOS, REGRAS E CONSULTAS

Em Prolog, uma base de dados é alimentada com cláusulas (i.e., *fatos* e *regras*). Sob esta podem ser realizadas *consultas*. Como veremos, estes elementos básicos do Prolog são muito próximos do que vimos em lógica de predicados.

Fatos

Podendo ser usados para definir *fatos*, os predicados são a unidade básica de Prolog e são dados por um identificador acompanhado de um ou mais argumentos. Por exemplo,

```
homem(sócrates) .
```

informa o *fato* de que Sócrates é um homem ao declarar o predicado `homem` com um argumento, o átomo `sócrates`. De maneira análoga, podemos usar predicados para definir *fatos* que estabelecem relacionamentos. Por exemplo,

```
pai(adão, cain) .
```

estabelece que Adão é pai de Cain ao declarar, através do predicado `pai`, que a relação `pai` existe entre `adão` e `cain`.

Observe que, em Prolog, identificadores de predicados e de átomos são sempre representados por palavras iniciadas com letras minúsculas.

O Exemplo 2.1 mostra a sequência de declarações como parte da construção de uma base de dados para representar uma determinada árvore genealógica.

Exemplo 2.1. Árvore genealógica

```
pai(adão,cain) .  
pai(adão,abel) .  
pai(adão,seth) .  
pai(seth,enos) .
```

Consultas

Uma vez que temos uma base de dados estabelecida, consultas podem ser realizadas no *prompt* (?) do interpretador Prolog. Por exemplo, a consulta

```
?- pai(adão,cain) .
```

verifica, junto à base de dados, se a relação `pai` está estabelecida entre os objetos `adão` e `cain`, em outras palavras, se Adão é pai de Cain. Se considerarmos a base de dados construída por meio do Exemplo 2.1, a consulta acima geraria a resposta `yes` do interpretador Prolog. Note que, responder (`yes` ou `no`) à uma determinada consulta consiste de determinar computacionalmente se uma proposição é consequência lógica dos fatos (e regras) estabelecidos na base de dados.

Variáveis

Em Prolog, variáveis são identificadas por palavras iniciadas com letras maiúsculas. Elas podem ser usadas em consultas para se buscar na base de dados os possíveis valores de argumento em um predicado, que o faz verdadeiro. Por exemplo, a seguinte consulta

```
?- pai(X,abel) .
```

representa a pergunta “*quem é pai de Abel?*”. Em outras palavras, ela busca pelos valores de `X` que tornam a consulta pela relação `pai` entre `X` e `abel` uma consequência lógica da base construída. A resposta à esta consulta, tendo em vista o Exemplo 2.1, seria `X = adão`.

Caso o sejam encontradas mais de uma resposta o sistema apresentará a primeira resposta encontrada. Ao pressionar a tecla *Enter* o usuário informa ao sistema para encerrar a consulta, ao pressionar a tecla *ponto e vírgula (;)* o usuário informa ao sistema para buscar por mais um resultado. Por exemplo,

```
?- pai(adão,X) .  
X = cain ;  
X = abel ;  
X = seth ;  
no
```

Variáveis Compartilhadas

Ao realizar uma consulta, podemos estabelecer uma variável e compartilhá-la entre predicados distintos. Por exemplo, suponha que desejamos realizar uma consulta pelo avô de Enos. Observe que a relação `avô` não foi estabelecida no Exemplo 2.1. Ainda assim poderíamos realizar a seguinte pergunta, que faz uso da relação `pai`, “*quem é o pai do pai de Enos?*”. Na notação Prolog a consulta ficaria como segue

```
?- pai(Y,enos), pai(X,Y).
```

Note que num primeiro momento será realizada uma busca pelos objetos Y que satisfazem a relação $\text{pai}(Y, \text{enos})$. Posteriormente, substituindo os resultados de Y em $\text{pai}(X, Y)$, buscam-se valores para X . A resposta do interpretador Prolog para esta consulta, tendo em vista a base de dados do Exemplo 2.1, seria

```
Y = seth
X = adão
yes
```

Variáveis Anônimas

As variáveis anônimas (representadas pelo símbolo *underscore* ‘_’) são usadas quando, ao realizar uma consulta, a resposta a um determinado argumento for irrelevante. Por exemplo, se quisermos buscar pelos elementos que são pais de algum outro elemento na base de dados, poderíamos realizar a seguinte consulta

```
?- pai(X, _).
```

E tendo em vista a base de dados do Exemplo 2.1 obteríamos a seguinte resposta

```
X = adão
X = seth
```

Regras

As regras consistem de outra maneira de representar predicados para alimentar as bases de dados em Prolog. Elas nos permitem definir predicados em termos de outros predicados. Por exemplo, a regra

```
avô(X,Y) :- pai(X,Z), pai(Z,Y).
```

assim como discutido na Seção 0, estabelece que a relação de avô pode ser obtida por meio da relação pai, ou seja, se X é pai de Z e Z é pai de Y então X é avô de Y . Portanto, uma vez que esta regra é adicionada à base de dados do Exemplo 2.1, a seguinte consulta poderá ser realizada

```
?- avô(X,enos)
X = adão
```

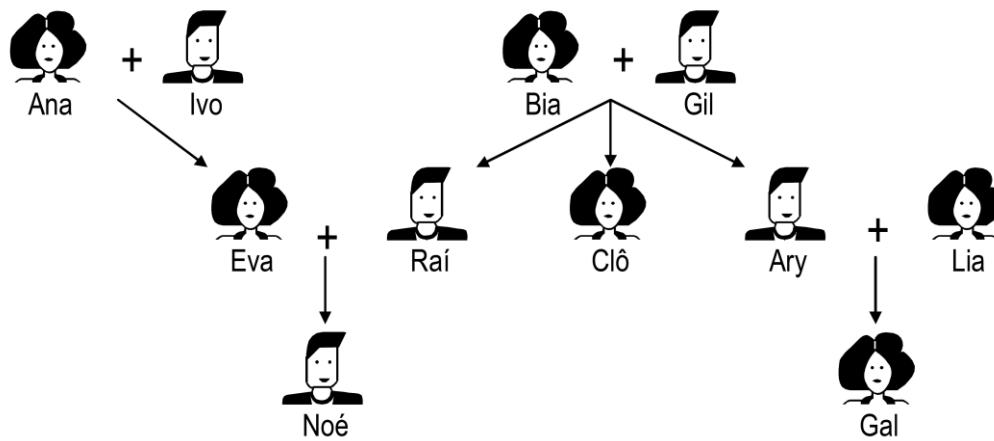
Um outro exemplo de relação genealógica que pode ser estabelecido por meio de regras pai consiste da relação irmão. A regra

```
irmão(X,Y) :- pai(Z,X), pai(Z,Y), X\=Y
```

estabelece que X é irmão de Y se tanto X quanto Y tiverem o mesmo pai (no caso Z) e X e Y não forem o mesmo elemento.

EXERCÍCIOS

- 1) Rediga a base de dados do Exemplo 2.1 e acrescente as relações avô e irmão. Realize as seguintes consultas
 - a) Quem são os filhos de Adão?
 - b) Quem são os netos de Adão?
 - c) Quem são os tios de Enos?
- 2) Considere a árvore genealógica abaixo:



- a) Usando fatos, defina as relações pai e mãe.
- b) Acrescente fatos que estabelecem quem é homem e quem é mulher.
- c) Usando regras, defina a relação gerou(X,Y) que determina se X é pai ou mãe de Y.
- d) Usando as relações já existentes defina regras para as relações filho, filha, tio, tia, primo, prima, avô e avó.