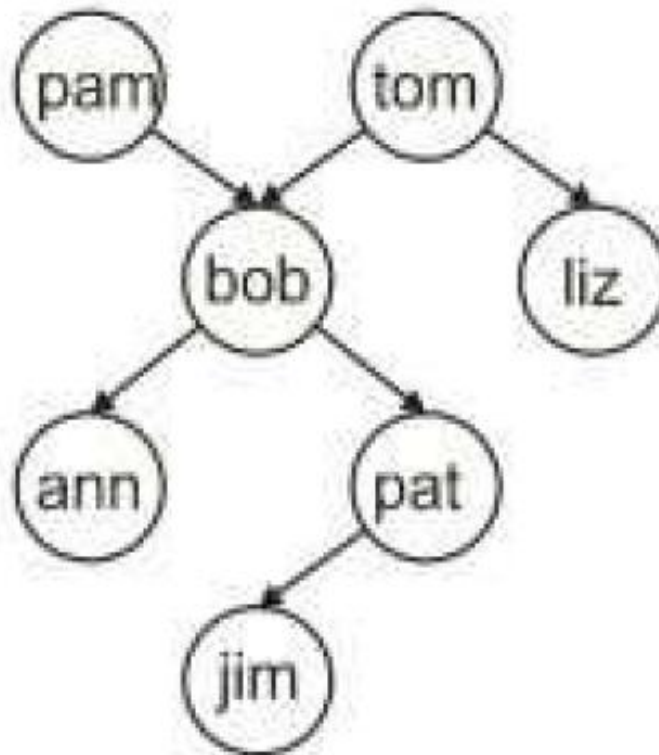


Linguagem de Programação Prolog

Prof. Dr. Eduardo Takeo Ueda
edutakeo@gmail.com

Árvore genealógica (1/4)



Árvore genealógica (2/4)

■ Fatos

```
genitor(pam, bob).  
genitor(tom, bob).  
genitor(tom, liz).  
genitor(bob, ann).  
genitor(bob, pat).  
genitor(pat, jim).
```

■ Consultas

```
3 ?- genitor(pat, jim).  
Yes  
4 ?- genitor(jim, pat).  
No
```

Árvore genealógica (3/4)

5 ?- genitor(X, bob).

X = pam ;

X = tom ;

No

6 ?- genitor(bob, X).

X = ann ;

X = pat ;

No

7 ?- genitor(X,Y).

X = pam

Y = bob ;

X = tom

Y = bob ;

X = tom

Y = liz ;

X = bob

Y = ann ;

X = bob

Y = pat ;

X = pat

Y = jim ;

No

Árvore genealógica (4/4)

- Fato não descrito diretamente
- Quem são os avós?

```
8 ?- genitor(Y, jim) , genitor(X,Y).  
Y = pat  
X = bob ;  
No
```

Predicados homem(X) e mulher(X)

- homem(X) significa “X é homem”
- mulher(X) significa “X é mulher”

```
mulher(pam).  
homem(tom).  
homem(bob).  
mulher(liz).  
mulher(pat).  
mulher(ann).  
homem(jim).
```

Retornando para a árvore genealógica

- Quem é a mãe de bob?

```
9 ?- genitor(X, bob), mulher(X).  
X = pam ;  
No
```

- Consulta com negação

```
10 ?- genitor(X, bob), not(homem(X)).  
X = pam ;  
No
```

Resolução e Unificação

- **Resolução:** é uma regra de inferência que permite que proposições inferidas sejam computadas a partir de proposições dadas
- **Instanciação:** atribuição de valores a variáveis durante a resolução
- **Unificação:** é um processo de correspondência de padrões que determina que instâncias podem ser feitas a variáveis ao mesmo tempo que faz várias resoluções

Backtracking

Uma busca inicial em um programa em Prolog segue o padrão **busca em profundidade** (*depth-first search*), ou seja, a árvore é percorrida sistematicamente de cima para baixo e da esquerda para direita. Quando essa pesquisa falha, ou é encontrado um nó terminal da árvore, entra em funcionamento o mecanismo de **backtracking**. Esse procedimento faz com que o sistema retorne pelo mesmo caminho percorrido com a finalidade de encontrar soluções alternativas.

Backtracking - Exemplo

- ?- pai(alexandre,X), mae(helena,X)
- Prolog tenta satisfazer a primeira sentença objetivo (proposição). Quando conseguir, tenta satisfazer a segunda. Caso não consiga, ele retorna ao ponto onde encontrou a solução para o primeiro objetivo (*backtracking*).

Cut (Corte)

- O comando **cut** permite indicar ao Prolog quais sub-objetivos já satisfeitos não necessitam ser reconsiderados ao se realizar um *backtracking*.
- O uso do comando **cut** é importante porque permite que um programa Prolog execute mais rápido.
- Em muitas situações o comando **cut** evita que um programa Prolog entre em um *laço infinito*.

Cut (Corte) - Exemplo

- O comando **cut** é representado pelo símbolo !
- `primogenito(X,Y) :- pai(Y,X), masculino(X), !`
- O código acima faz a consulta a base de conhecimento e para na primeira ocorrência de filho do sexo masculino.

Cut (Corte) - Exemplo

- O comando **cut** é representado pelo símbolo !
- `primogenito(X,Y) :- pai(Y,X), masculino(X), !`
- O código acima faz a consulta a base de conhecimento e para na primeira ocorrência de filho do sexo masculino.

Cut (Corte) - Aplicações

- Unificação de padrões, de forma que quando um padrão é encontrado os outros padrões possíveis são descartados.
- Eliminação da árvore de pesquisa de soluções alternativas quando apenas uma é suficiente.
- Encerramento da pesquisa quando a continuação iria conduzir a uma pesquisa infinita.

Comandos úteis no SWI-Prolog

- **listing**
- **trace** (notrace)
- **debug** (nodebug)
- **quit**
- **help**

Atividade Discente Orientada

(1) Retorne ao exemplo da árvore genealógica e escreva **regras** para determinar mãe, avós e irmã.

(2) Reescreva em **Prolog** todos os exemplos (em Haskell) aprendidos nesta disciplina