Universidade Federal da Bahia
Instituto de Computação – DCC/UFBA

Aula 06 – PROLOG – Buscas e Consultas

Disciplina: MATA56 - Paradigmas de Linguagens de Programação

Profº Claudio Junior Nascimento da Silva

claudiojns@ufba.br

Roteiro

- Consultas e Buscas: teoria
- Exercícios em sala:
  - Exercício 09 – Unificação dos termos;
  - Exercício 10 – Verificar consultas - I;
  - Exercício 11 – Verificar consultas – II.
- Lista de Exercícios I – Correção/dúvidas.

Prolog - Prática

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

Memória...

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]

Consultas

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.

Unificação

- Ao responder a uma consulta, o interpretador Prolog pode precisar instanciar variáveis, isto é, atribuir valor às variáveis;
- Dois termos podem ser unificados se:
  1. Eles são iguais;
  2. Contêm variáveis que podem ser instanciadas de forma a tornar os termos idênticos;
- Ou seja:
  - Dois termos podem ser unificados se, e somente se uma das condições for verdadeira:
    - Os dois termos são constantes (átomos ou números) e iguais:
      - ✓ 1 e 1, casa e casa, joao e joao.
    - Um dos termos é uma variável:
      - ✓ X e 1, X e masculino(harry), X e Y.
    - Ambos são termos complexos como o mesmo nome e mesmo número de argumentos, e os termos correspondentes podem ser unificados:
      - ✓ pai(james, harry) e pai(X, harry)

Unificação

- O predicado = pode ser usado para determinar se dois termos podem ser unificados:
  - =(casa, casa). ou casa = casa.
  - X = casa.
  - X = casa, Y = X
  - X = Y
  - Negação: \=. \=(X, Y) é equivalente a not(=(X,Y)).

Exercício 09

- Quais dos termos pares abaixo é possível unificar?

Termos	Resultado
bread = bread	
'Bread' = bread	
'bread' = bread	
Bread = bread	
bread = sausage	
food(bread) = bread	
food(bread) = X	

Termos	Resultado
food(X) = food(bread)	
food(bread,X) = food(Y,sausage)	
food(bread,X,beer) = food(Y,sausage,X)	
food(bread,X,beer) = food(Y,kahuna_burger)	
food(X) = X	
meal(food(bread),drink(beer)) = meal(X,Y)	
meal(food(bread),X) = meal(X,drink(beer))	

1 – Termos iguais
2 – Contém variáveis que podem ser iniciadas

A. Os dois termos são átomos ou números e iguais
B. Um dos termos é uma variável
C. São termos complexos, com mesmo nome e número de argumentos e os termos correspondentes podem ser unificados

Busca

- Ao receber uma consulta, o Prolog tenta unifica-la com cada um dos fatos e com a cabeça das regras que estão na base de conhecimento:
  - Lembrete: regra = cabeça :- corpo.
- Se unificar com um fato, retorna as instâncias das variáveis;
- Se unificar com a cabeça de uma regra, consulta o corpo da regra (recursivamente).

Busca - Exemplo

```
%fatos
f(a).
f(b).

g(a).
g(b).

h(b).

%regras
k(X) :- f(X) , g(X) , h(X).
```

- Qual o resultado da pesquisa: ?- k(Y).
  - *k(b).*
- Como funciona?
  1. Identifica a estrutura da consulta:
     1. Predicado: k;
     2. Argumento: Y.
  2. Leitura da base;
  3. Existe algum Fato com a mesma estrutura da consulta?
  4. Existe alguma Regra com a mesma estrutura da consulta?
     1. k(X) :- f(X) , g(X), h(X).
     2. Y ➔ Variável
        1. k(Y) :- f(a) , g(a) , h(a) é verdadeiro?
        2. k(Y) :- f(b) , g(b) , h(b) é verdadeiro?
        3. k(Y) :- f(c) , g(c) , h(c) é verdadeiro?
     3. Qual o resultado?
     4. Existe alguma outra Regra?

Busca – Exemplo

```
% Fatos
homem(tom).
mulher(pam).

genitor(pam, bob).
genitor(tom, bob).


%Regras
```

```
?- genitor(tom, Y). % tom é genitor de quem (Y)?
```

- Existe Fato com a estrutura genitor(X,Y)?
  - genitor(tom,Y) = genitor(pam, bob)?
  - genitor(tom,Y) = genitor(tom, bob)?
  - Algum outo Fato?
- Existe Regra com a estrutura genitor(X,Y)?

```
?- genitor(X, bob). % Quem (X) é/são o(s) genitores de bob?
```

- Existe Fato com a estrutura genitor(X,Y)?
  - genitor(X, bob) = genitor(pam, bob)?
  - genitor(X, bob) = genitor(tom, bob)?
  - Algum outo Fato?
- Existe Regra com a estrutura genitor(X,Y)?

Exercício 10

- Utilizando a base de conhecimento abaixo, quais consultas são satisfeitas? Se necessário, faça as devidas correções.

```
%fatos
house_elf(dobby).
witch(hermione).
witch('McGonagall').
witch(rita_skeeter).

%regras
magic(X) :-  house_elf(X).
magic(X) :-  wizard(X).
magic(X) :-  witch(X).
```

Consulta	Resultado
?- magic(house_elf).	
?- wizard(harry).	
?- magic(wizard).	
?- magic('McGonagall').	
?- magic(Hermione).	

Exercício 11

- Utilizando a base de conhecimento abaixo, qual o retorno da consulta abaixo? Se necessário, faça as devidas correções.

```
%fatos
progenitor(paul, petunia).
progenitor(paul, lili).

%regras
tem_mesmo_progenitor_que(X, Y) :-
        progenitor(P, X) , progenitor(P, Y).

eh_irma_de(X, Y) :-
     X \== Y, tem_mesmo_progenitor_que(X,Y).
```

- Qual o retorno da consulta:
  - eh_irma_de(petunia, Y). ?
- Por que ?
- Como resolver o problema?