Aula 2: Tipo de dados, Match, Retrocesso

Tipo de dados - Termos

Átomo: uma sequência formada por letras maiúsculas, minúsculas, dígitos e underline que inicie com letra minúscula.

String: uma sequência de caracteres entre aspas. Em porlog, para fins de relação, uma string é um átomo.

Numérico: float e int.

Constante: são termos do tipo átomo, string e numérico.

Termo complexo: um termo complexo é formado por um predicado seguido pela espeficicação dos seus termos dentro de parênteses. O predicado, também chamado de functor, deve ser um átomo.

Variável: uma sequência formada por letras maiúsculas, minúsculas, dígitos e underline que inicie com letra maiúscula. A variável _ é uma variável especial, chamda de variável anônima.

Matching

Definição: Dois termos dão match, se são iguais ou se contêm variáveis que podem ser instanciadas de forma que os termos resultantes sejam iguais. Isso significa que:

Se t1 e t2 são constantes, então t1 e t2 dão match se, e somente se, são o mesmo átomo ou o mesmo número.

Se t1 é uma variável não instanciada e t2 é qualquer tipo de termo, então t1 é instanciada com o valor de t2 e t1 e t2 dão match.

Se t1 é uma variável instanciada e t2 é qualquer tipo de termo, t1 e t2 dão match se, e somente se, a instância da variável t1 for o mesmo valor t2.

Se t1 e t2 são termos complexos, eles dão match se, e somente se:

a- Eles têm o mesmo predicado.

b- Todos os seus argumentos correspondentes dão match.

Informe quais relações dão match

A = 2

In [ ]:
```
2 = '2'
```

In [ ]:
```
mia = Mia
```

In [ ]:
```
f = f
```

In [ ]:
```
X = Y
```

In [ ]:
```
f(h(t), K) = f(W, w(r))
```

In [ ]:
```
loves(X, X) = loves(mia, teo)
```

In [ ]:
```
h(X) = X
```

Exemplos de utilização de termos complexos

Exemplo: Programa para verificar se dois pontos formam uma linha horizontal.

In [ ]:
```
horizontal(line(point(X,Y),point(Z,Y))).
```

Exemplo: Programa para verificar se dois pontos formam uma linha vertical.

In [ ]:
```
vertical(line(point(X,Y),point(X,Z))).
```

Vejamos algumas consultas

In [ ]:
```
vertical(line(point(1,1),point(1,3))).
```

In [ ]:
```
vertical(line(point(1,1),point(3,2))).
```

In [ ]:
```
horizontal(line(point(2,3),P)).
```

Retrocesso

Ao realizar uma consulta, o interpretador prolog utiliza o algoritmo de backtracking para encontrar as respostas. A busca implica na construção de vários caminhos de uma árvore. Todos os caminhos bem sucedidos serão considerados para a resposta. Para analisar o processo de busca do interpretador, basta utilizar o predicado trace\0 no momento da consulta. Por exemplo: trace, irmão(ary, X).

Exemplo: Dado a base de conhecimento, qual o resultado para as consultas:

In [ ]:
```
f(a).
f(b).
```

? f(a)

? f(b)

Exemplo: Dado a base de conhecimento, qual o resultado para as consultas:

In [ ]:
```
f(a).
f(b).
f(X) :- X == a.
```

? f(a)

? f(b)

Exemplo: Dado a base de conhecimento, qual o resultado para as consultas:

In [ ]:
```
f(X, 0) :- X < 5.
f(X, 1) :- X >= 5, X =< 9.
f(X, 2) :- X > 9.
```

? f(4, R)

Exemplo: Dado a base de conhecimento, qual o resultado para a consulta:

In [ ]:
```
d(0).
d(1).
bin :- d(A), d(B), d(C), write([A,B,C]), nl, fail.
```

? bin

Corte

O operador ! (cut) é utilizado par podar a ramificação da busca do interpretador prolog. Normalmente, é utilizado para evitar retrocesso desnecessário. O operador cut também pode ser utilizado para realizar controle procedimental.

Evitando retroceso desnecessário

In [ ]:
```
f(X, 0) :- X < 5, !.
f(X, 1) :- X >= 5, X =< 9, !.
f(X, 2) :- X > 9.
```

Realizando controle procedimental

In [ ]:
```
if(Condition,Then,_) :- Condition, Then, !.
if(_,_,Else) :- Else.
```

Falha

Uma forma de forçar o retrocesso em busca de soluções alternativas, sem que o usuário tenha que solicitar, é fazer com que após cada resposta obtida o sistema encontre um objetivo insatisfatível. Em Prolog, esse objetivo é representado pelo predicado fail, cuja execução sempre provoca uma falha.

In [ ]:
```
d(0).
d(1).
bin :- d(A), d(B), d(C), write([A,B,C]), nl, fail.
```

Comparação e unificação de termos

=/2

Unifica o termo 1 e o termo 2. True, se a unificação for bem sucedida.

In [ ]:
```
a = a
b = 'b'
A = a
B = 5
C = B + 1
D = 6
```

\=/2

Equivalente a \+ termo1 = termo2

Se os termos estiverem devidamente instanciados, verifica a não equivalência das instâncias.

Comparação de termos

termo 1 == termo 2

True ser termo 1 é equivalente ao termo 2.

termo 1 \== termo 2

True ser termo 1 não é equivalente ao termo 2.

termo 1 @< termo 2

True se o termo 1 precede o termo 2 na ordem de precedência. Ordem de precedência: variável < numérico < string < átomo < termo complexo

termo 1 @=< termo 2

True se o termo 1 é igual ao termo 2 ou se o termo 1 precede o termo 2 na ordem de precedência.

termo 1 @> termo 2

True se o termo 1 sucede termo 2 na ordem de precedência.

termo 1 @=> termo 2

True se o termo 1 é igual ao termo 2 na ordem de precedência ou se o termo 1 sucede termo 2 na ordem de precedência.

Operações aritméticas

Prolog oferece um predicado especial is, bem como um conjunto operadores, através dos quais podemos efetuar operações aritméticas.

Operadores artiméticos

Os operadores aritméticos são + (adição), - (subtração), * (multiplicação), mod (resto), / (divisão real), // (divisão inteira) e ^ (potenciação).

Operador is: Número is Expr

True, quando o valor do lado esquerdo é igual ao resultado do processamento do lado direito. Se no lado esquerdo for utilizada uma variável não instanciada, então ela será instanciada com o resultado do processamento da expressão.

In [ ]:
```
X is 3 + 4
```

In [ ]:
```
Y is X * 2
```

Exemplo

In [ ]:
```
% país(Nome, Área, População)
país(brasil, 9, 130).
país(china, 12, 1800).
país(eua, 9, 230).
país(índia, 3, 450).
```

A base de conhecimento representa uma tabela que relaciona a cada país sua área em Km2 e sua população em milhões de habitantes. Com base nesse programa, podemos determinar a densidade demográfica do Brasil, através da seguinte consulta:

In [ ]:
```
país(brasil,A,P), D is P/A.
```

Qual a diferença entre a população da chine e do brasil?

In [ ]:
```
país(chine,_,Pc), país(brasil,_,Pb), D is Pc - Pb
```

Operadores relacionais

Para realizar comparações numéricas podemos usar os seguintes predicados primitivos: =:= (igual) , =\= (diferente), >

(maior), >= (maior ou igual), < (menor) e =< (menor ou igual).

A área do Brasil é igual à área dos Estados Unidos?

In [ ]:
```
país(brasil,Ab,_), país(eua,Ae,_), Ab =:= Ae
```

Exercício

1- Considerando a base de conhecimento:

In [ ]:
```
%func(Código, Nome, Salário)
func(1, ana, 1000.90).
func(2, bia, 1200.00).
func(3, ivo, 903.50).

% dep(Código, Nome)
dep(1, ary).
dep(3, raí).
dep(3, eva).
```

a) Quais os dependentes do Ivo?

b) De quem Ary é dependente?

c) Quem depende de funcionário com salário inferior a R$ 950,00?

d) Quais funcionários que não têm dependentes?

2- Considerando a base de conhecimento:

In [ ]:
```
joga(ana,volei).
joga(bia,tenis).
joga(ivo,basquete).
joga(eva,volei).
joga(leo,tenis).
```

Suponha que desejamos consultar esse programa para encontrar um parceiro P para jogar com Leo. Então, podemos realizar essa consulta de duas formas:

In [ ]:
```
joga(leo,X), joga(P,X), P \== leo
joga(P,X), joga(leo,X), P \== leo
```

Qual consulta é mais eficiente, por quê?

3- O predicado num classifica números em três categorias: positivos, nulo e negativos. Esse predicado, da maneira como está definido, realiza retro- cesso desnecessário. Explique por que isso acontece e, em seguida, utilize cortes para eliminar esse retrocesso.

In [ ]:
```
num(N,positivo) :- N>0.
num(0,nulo).
num(N,negativo) :- N<0.
```

4- Considerando os dados:

In [ ]:
```
nomes = ['Ana', 'Bia', 'Ivo', 'Lia', 'Eva', 'Ary']
sexo = [fem, fem, masc, fem, fem, masc]
idade = [23, 19, 22, 17, 28, 25]
altura = [1.55, 1.71, 1.80, 1.85, 1.75, 1.72]
peso = [56.0, 61.3, 70.5, 57.3, 68.7, 68.9]
```

Construa uma base de conhecimento com o predicado cadastro/5 e realiza as seguintes consultas:

a) Quais são as mulheres com mais de 20 anos de idade?

b) Quem tem pelo menos 1.70m de altura e menos de 65kg?

c) Quais são os possíveis casais onde o homem é mais alto que a mulher?

5- O peso ideal para uma modelo é no máximo 62.1*Altura–44.7 . Além disso, para ser modelo, uma mulher precisa ter mais que 1.70m de altu- ra e menos de 25 anos de idade. Com base nessas informações, e conside- rando a tabela do exercício anterior, defina um predicado capaz de recu- perar

apenas os nomes das mulheres que podem ser modelos.