

Prolog e Programação Lógica

PARADIGMAS DE PROGRAMAÇÃO

Antes da programação lógica...

- Antes de falar de programação lógica, vamos falar rapidamente das tuplas:
- Uma tupla é um agrupamento de valores sem nome, mas ordenados, possivelmente de tipos diferentes.

Tuplas

```
(1, 2)
```

```
// 3 strings.  
("um", "dois", "tres")
```

```
// tipos genéricos.  
(a, b)
```

```
// tipos diferentes.  
("one", 1, 2.0)
```

```
// expressões de inteiros.  
(a + 1, b + 1)
```

Tuplas

- As tuplas podem ter seus membros acessados a partir do match:

```
let print tupla1 =  
  match tupla1 with  
  | (a, b) -> printfn "Par %A %A" a b
```

Tuplas

- Podemos criar valores a partir de tuplas, valores indesejados podem ser substituídos por `_`.

```
let (a, b) = (1, 2)
let (a, _) = (1, 2)
```

```
let c = fst (1, 2)
let d = snd (1, 2)
```

- Existem métodos para resgatar o primeiro e segundo valor de uma tupla (`fst` e `snd` respectivamente), para o terceiro em diante é preciso criar manualmente, como o exemplo do terceiro a seguir.

```
let terceiro (_, _, c) = c
```

Tuplas

- As tuplas fornecem uma maneira conveniente de retornar vários valores de uma função, como pode ser utilizado para a criação do próximo passo do fibonacci por pipeline.

Prolog e Programação Lógica

PARADIGMAS DE PROGRAMAÇÃO

Prolog

- Prolog (Programação Lógica) é uma linguagem de programação que se enquadra no paradigma de Programação em Lógica Matemática. É uma linguagem de uso geral que é especialmente associada com a inteligência artificial e linguística computacional. Consiste numa linguagem puramente lógica, que pode ser chamada de Prolog puro, e numa linguagem concreta, a qual acrescenta o Prolog puro com componentes extra-lógicos.

Prolog x Outras Linguagens

- **Linguagens Procedimentais (C, Pascal, Basic...):** Especifica-se como realizar determinada tarefa.
- **Linguagens Orientadas a Objetos (C++, Java, C#...):** Especifica-se objetos e seus métodos.
- **Prolog:** Especifica-se o quê se sabe sobre um problema e o quê deve ser feito. É mais direcionada ao conhecimento e menos direcionada a algoritmos

Prolog

- Programar em Prolog envolve:
 - Declarar alguns fatos a respeito de objetos e seus relacionamentos.
 - Definir algumas regras sobre os objetos e seus relacionamentos.
 - Fazer perguntas sobre os objetos e seus relacionamentos.

Sentenças

- Nomes de constantes e predicados iniciam sempre com letra minúscula.
- O predicado (relação unária, n-ária ou função) é escrito primeiro e os objetos relacionados são escritos depois entre parênteses.
- Variáveis sempre começam por letra maiúscula.
- Toda sentença termina com ponto “.”
- Exemplo: gosta(maria, uva).

SWI - Prolog

- Existem algumas implementações do Prolog, vamos utilizar por enquanto o SWI-Prolog que é open-source e tem uma opção para trabalharmos on-line:
 - <https://swish.swi-prolog.org/>

Prolog

- Ao criar um novo documento no SWISH, a tela é dividida em três partes:
 - Esquerda: base de conhecimento (programa)
 - Direita inferior: consulta
 - Direita superior: resultados

Exemplo Inicial

- Exemplo inicial
- Escreva, na base de conhecimento, o seguinte:
 - humano(joao).
- Com isso, você está afirmando que joao é humano. humano(joao), nesse contexto, é um fato.
- Agora, na área de consulta, escreva o seguinte:
 - humano(joao).
- Você está fazendo uma consulta: joao é humano? Na área de resultados, o interpretador vai responder true.

Exemplo Inicial

- Outras consultas para testar:
 - humano(shrek).
 - humano(caetano).
 - humano(X).
 - false significa “não é possível provar que é verdadeiro” (hipótese do mundo fechado).

Exemplo maior

- Vamos tentar um exemplo mais completo. Digite o seguinte na base de conhecimento:
 - animal(tomba).
 - animal(rex).
 - animal(bolota).
 - cachorro(tomba).
 - cachorro(rex).
 - gato(bolota).

Exemplo maior

- Agora digite a seguinte consulta:
 - `animal(X).`
- Nesse contexto, `X` é uma variável. O interpretador Prolog vai retornar todos os valores de `X` para os quais `animal(X)` é verdadeiro. Nesse caso, o resultado vai ser `X = tomba`, `X = rex` e `X = bolota`.

Exemplo maior

- Se sabemos que rex é um cachorro, não deveríamos precisar dizer que ele é um animal, pois todo cachorro é um animal! A mesma lógica vale para os gatos! Podemos inserir regras na nossa base de conhecimento:
 - - `animal(X) :- gato(X).`
 - `animal(X) :- cachorro(X).`
 - `gato(bolota).`
 - `cachorro(tomba).`
 - `cachorro(rex).`
 - As duas primeiras linhas são regras, que podem ser traduzidas como:
 - Se X é gato, então X é um animal.
 - Se X é cachorro, então X é um animal.
 - Agora realize a consulta `animal(rex)`. Qual é o retorno?
- Realize também a consulta `animal(X)`. Qual é o retorno?

Operadores Lógicos

Simbolo	Conectivo	Opração Lógica
:-	IF	Implicação
,	AND	Conjunção
;	OR	Disjunção
not	NOT	Negação

Operadores Relacionais

Operador	Significado
$X = Y$	Igual a
$X \neq Y$	Não igual a
$X < Y$	Menor que
$Y > X$	Maior que
$Y \leq X$	Menor ou igual a
$Y \geq X$	Maior ou igual a

Conceitos

- Conceitos
- Um programa em Prolog é uma base de conhecimento que contém cláusulas finalizadas por ponto (.). Essas cláusulas representam o que sabemos sobre o mundo.
- Uma cláusula pode ser um fato ou uma regra. Uma regra possui cabeça e corpo, no formato cabeça :- corpo. A interpretação da regra é “se o corpo é verdadeiro, então a cabeça é verdadeira”.
- Identificadores começados por letras minúsculas são átomos, que representam objetos específicos do nosso mundo. Identificadores começados por letras maiúsculas são variáveis.

Conceitos

- Uma variável tem validade apenas dentro da cláusula onde se encontra. Assim, duas ocorrências de `X` na mesma cláusula correspondem à mesma variável:
 - % Se `X` for `joao` e `pensa(joao)` é verdadeiro,
 - % concluímos que `existe(joao)`.
 - `existe(X) :- pensa(X)`.
- Por outro lado, é possível usar o mesmo nome em cláusulas diferentes para se referir a coisas diferentes:
 - % Podemos ter `X = rex` na primeira cláusula
 - % e `X = bolota` na segunda cláusula.
 - `animal(X) :- cachorro(X)`.
 - `animal(X) :- gato(X)`.

Exercícios

- Fatos e consultas
- Considere a seguinte base de conhecimento composta de predicados progenitor/2:
 - progenitor(maria, jose).
 - progenitor(joao, jose).
 - progenitor(joao, ana).
 - progenitor(jose, julia).
 - progenitor(jose, iris).
 - progenitor(iris, jorge).
- Considere que o predicado progenitor(A, B) significa que A é progenitor (i.e., pai ou mãe) de B

Exercícios

1. Desenhe a árvore genealógica representada pela base de conhecimento.
2. Escreva uma consulta para responder à seguinte pergunta: “Ana é progenitora de Jorge?”
3. Escreva uma consulta para retornar os progenitores de Íris.
4. Escreva uma consulta para retornar os progenitores de José.
5. Escreva uma consulta para retornar todos os pares progenitor/filho da base de conhecimento.
6. Escreva uma consulta para retornar todos os avós de Jorge. Dica: sua consulta será formada por dois termos separados por vírgula.
7. Escreva uma consulta para retornar todos os netos de João.
8. Escreva uma consulta para retornar todos os progenitores comuns de José e Ana.