

Trabalhando com o Paradigma Declarativo

Prof. Rodrigo Hübner

Resumo

Este trabalho prático tem por objetivo a fixação e implementação de programas em Prolog e utilizar tais programas como parte integrante de um serviço web para aquisição de dados.

1. Introdução

Prolog é uma linguagem declarativa, mais utilizada hoje entre as linguagens do paradigma lógico.

Este trabalho tem como objetivo conhecer e aprender novas práticas da programação lógica (ou declarativa), sugerindo a implementação de alguns exercícios simples de fixação e alguns mais complicados para explorar a linguagem. Além disso, tais exercícios servirão para a aquisição de dados em um *webservice* construído com a linguagem de programação **Racket**.

As próximas seções descrevem (2) o trabalho proposto, (3) a avaliação, (4) questões importantes e, por fim, (5) a conclusão.

2. O Trabalho

Exercícios de fixação:

1. Expresse através de fatos e regras Prolog as informações contidas na seguinte frase: “João é um pássaro. Pedro é um peixe. Maria é uma minhoca. Pássaros gostam de minhocas. Gatos gostam de peixes. Gatos gostam de pássaros. Amigos gostam uns dos outros. O meu gato é meu amigo. O meu gato come tudo o que gosta, exceto pessoas. O nome do meu gato é Chuck Norris.”
2. Expresse através de fatos e regras Prolog as informações contidas na seguinte frase: “Cassia é bonita. Marcos é rico e bonito. Ana é rica e forte. Fabiano é forte e bonito. Silvio é amável e forte. Todos os homens gostam de mulheres bonitas. Todos os homens ricos são felizes. Qualquer homem

que gosta de uma mulher que gosta dele é feliz. Qualquer mulher que gosta de um homem que gosta dela é feliz. Ana gosta de qualquer homem que gosta dela. Cassia gosta de qualquer homem que gosta dela, desde que ele seja rico, amável ou bonito, e forte.”

3. Considere a seguinte base de fatos em Prolog:

```
aluno(joao, calculo).
aluno(maria, calculo).
aluno(joel, programacao).
aluno(joao, estrutura).
frequenta(joao, uem).
frequenta(maria, uem).
frequenta(joel, utfpr).
professor(carlos, calculo).
professor(ana_paula, estrutura).
professor(pedro, programacao).
funcionario(pedro, utfpr).
funcionario(ana_paula, uem).
funcionario(carlos, uem).
```

Escreva as seguintes sentenças em Prolog:

- a) Quem são os alunos do professor X?
- b) Quem são as pessoas que estão associadas a uma universidade X?
(alunos e professores).

4. Suponhas os seguintes fatos:

```
nota(joao, 5.0).
nota(maria, 6.0).
nota(joana, 8.0).
nota(mariana, 9.0).
nota(cleuza, 8.5).
nota(jose, 6.5).
nota(joaquim, 4.5).
nota(mara, 4.0).
nota(mary, 10.0).
```

Considerando que: Nota de 7.0 à 10.0 = Aprovado
Nota de 5.0 à 6.9 = Recuperação
Nota de 0.0 à 4.9 = Reprovado.

Escreva uma regra para identificar a situação de um determinado aluno.

5. Crie uma base de conhecimento em Prolog declarando os fatos representados na seguinte Tabela 1

5.1 Escreva regras genéricas em Prolog que possam responder as seguintes perguntas:

- a) Quem dirigiu o filme Titanic?
- b) Quais são os filmes de suspense?
- c) Quais os filmes dirigidos por Donner?
- d) Em que ano foi lançado o filme Sinais?
- e) Quais os filmes com duração inferior a 100min?
- f) Quais os filmes lançados entre 2000 e 2005?

5.2 Usando as regras criadas anteriormente construa o predicado “clássico”, que retorna o título dos filmes lançados antes de 1980.

5.3 Usando as regras criadas anteriormente construa o predicado “gênero”, que retorna o título dos filmes de um gênero específico.

5.4 Usando os predicados `clássico` e `gênero`, faça uma consulta para recuperar os títulos de filmes clássicos de suspense.

Tabela 1.

Catálogo de Filmes				
Título	Gênero	Diretor	Ano	Min.
Amnésia	Suspense	Nolan	2000	113
Babel	Drama	Inarritu	2006	142
Capote	Drama	Miller	2005	98
Casablanca	Romance	Curtiz	1942	102
Matrix	Ficção	Wachowsk	1999	136
Rebecca	Suspense	Hitchcock	1940	130
Shrek	Aventura	Adamson	2001	90
Sinais	Ficção	Shymalan	2002	106
Spartacus	Ação	Kubrik	1960	184
Superman	Aventura	Donner	1978	143
Titanic	Romance	Cameron	1997	194
Tubarão	Suspense	Spielberg	1975	124
Volver	Drama	Almodóvar	2006	121

Aplicações:

1. Construa um predicado que encontre o valor máximo entre duas listas
2. Construa um predicado para criar uma lista com a **intersecção** de duas outras listas.
3. Construa um predicado para criar uma lista com a **união** de duas outras listas.
4. Construa um predicado para calcular o fatorial de um número N: *factorial(N, Valor)*
5. Construa um predicado para calcular o fibonacci de um número N.

Junção para aquisição de dados com o *webservice* construído em Racket

Da mesma forma em que foi realizado em linguagem funcional, este trabalho agora tem como objetivo executar uma função **Prolog** por meio de um *webservice*.

A função que será executada e os parâmetros dela será obtidos por meio da URL pelo servidor escrito em Racket. A função correta deverá ser executada externamente em um compilador Prolog e a pesquisa realizada (parâmetros) deverão ser consultados pelo mecanismo de inferência em Prolog.

3. Avaliação do Trabalho

O trabalho será avaliado de acordo com os seguintes critérios:

Codificação

Será avaliado a modularização e a elegância do código fonte, além de estar de acordo com os critérios que são discutidos em sala de aula.

Execução

Será avaliado se os programas fazem realmente o que foi solicitado.

4. Questões Importantes

- **Cópias:** Qualquer tipo de cópia (trabalhos de colegas, internet, etc) **anulará** o trabalho, seja porções de código ou simplesmente o trabalho completo;
- **Entrega:** Deverá ser entregue o código fonte do trabalho em um arquivo compactado, com o nome no moodle;
- **Data:** O trabalho deverá ser entregue no dia 07/12/2015;
- **Valor:** 1,5 pontos;

5. Conclusão

Este trabalho tem por objetivo fixar exercícios e implementar aplicações em uma linguagem de programação lógica. Também é esperado que o estudante consiga desenvolver um só projeto com a junção de diferentes linguagens e paradigmas de programação. Após a execução deste trabalho é esperado que cada aluno tenha

aprimorado suas habilidades no paradigma lógico e seja capaz de se envolver em projetos em linguagens deste modelo.

Referências

- [1] <http://www.swi-prolog.org/>
- [2] <http://www.dcc.fc.up.pt/~vsc/Yap/>
- [3] <http://www.gprolog.org/>