Paradigmas da Programação III

Paulo Jorge Azevedo
Departamento de Informática
pja@di.uminho.pt

Resumo

Disciplina de introdução à programação em lógica na licenciatura de Matemática e Ciências da Computação.

1 A Disciplina

1.1 Objectivos

O objectivo principal desta disciplina é de transmitir os conhecimentos associados com os fundamentos da programação em lógica recorrendo à linguagem de programação PROLOG.

2 Programa

O programa da disciplina está dividido em duas componentes: uma componente teórica e outro prática.

2.1 Componente Teórica

- **Introdução à Programação em Lógica**

  1. Primeiros conceitos, computação como dedução, noção de consequência lógica. Lógica de Horn.
  2. Noções de Lógica de primeira ordem. Definições de Linguagem
  3. Teoria dos Modelos: Consequência Lógica, interpretações e modelos, validade e implicação lógica.
  4. Forma clausal. Conversão, (Skolemização)

5. Teoria das Provas: axiomas, teoremas e noção de derivabilidade de frases lógicas. Definição de Prova, Sistemas de inferência, regras de inferência, provas por refutação.
6. A regra de inferência *Resolução*. Resolução em lógica de predicados. Unificação, Noção de unificador, O algoritmo de Unificação
7. Programação em Lógica e Resolução. Resolução SLD, Árvores SLD, Regras de Computação e de estratégia de Pesquisa (Backtracking). *Pruning* da pesquisa em árvores SLD usando o operador **cut**.
8. Negação em Programação em Lógica. Regra do mundo fechado (Closed World Assumption). Implementação. Resolução SLDNF.

- **Programação em Prolog**

1. Definição do sistema Prolog como uma possível implementação de um sistema de programação em lógica.
2. Programas Recursivos. Uso de listas, árvores binárias, aritmética, expressões simbólicas, etc
3. Estruturas de Dados Incompletas: Diferença de listas, Diferença de estruturas.
4. Predicados predefinidos para decomposição de termos. $Functor$, Arg, *Univ* (= ..)
5. Predicados Meta-Lógicos: var, $nonvar$, $assert$, $retract$, $findall$, etc
6. Programação Não Determinística: *Generate and test*, tipos de não determinismo: *don't care* versus *don't know*. Jogos como exemplos de programação não determinística.
7. Meta-Interpretadores. Exemplos.
8. Métodos de transformação de programas: *Partial Evaluation*, *Folding and Unfolding*.

2.2 Componente Prática

Na componente prática serão resolvidos vários exercícios que consideram os temas desenvolvidos anteriormente na componente teórica.

3 Avaliação

O processo de avaliação envolverá duas vertentes: a realização de uma prova escrita e a realização de um trabalho prático. A nota final a atribuir a cada aluno será calculada pela seguinte formula:

Nota final = Trabalho pratico * 0.4 + Prova escrita * 0.6

São condições necessárias (mas não suficientes) para um aluno ser considerado aprovado:

- ter nota minima de 8 valores na prova teórica,
- e nota minima de 10 valores no trabalho prático.

Não haverá qualquer tipo de exame oral para os alunos com nota final não inferior a oito valores.

Referências

[Azevedo98] Azevedo P.
Paradigmas da Programação III - Notas teóricas
Universidade do Minho, 1998 - 2003.

[Hogger90] Hogger C.
Essentials of Logic Programming
Oxford University Press, 1990.

[Sterling&Shapiro86] Sterling L., Shapiro E.
The Art of Prolog
MIT Press, 1986.

[Bratko93] Bratko I.
Prolog Programming for Artificial Intelligence
Addison-Wesley, 1993