

Processamento de Linguagens e Compiladores

João Saraiva

jas@di.uminho.pt mailto:jas@di.uminho.pt

27 de Fevereiro de 2006

Esta página contém informação relativa à cadeira de *Processamento de Linguagens e Compiladores* a decorrer no *ano lectivo 2005-2006*. Processamento de Linguagens e Compiladores é uma cadeira do 2º Semestre do 2º Ano da licenciatura em Matemática e Ciências da Computação.

Esta página está a ser gerada a partir de um documento XML, por uma ferramenta que utiliza os métodos de programação leccionados neste curso (eg, expressões regulares, autómatos finitos e gramáticas independentes do contexto). A partir da especificação XML é produzida esta página, bem como a sua representação em LaTeX. O formato LaTeX dá origem aos seguintes documentos em postscript e pdf: [ps] http://www.di.uminho.pt/jas/Teaching/Courses/plc/05-06/pagina_plc.pdf pdf http://www.di.uminho.pt/jas/Teaching/Courses/plc/05-06/pagina_plc.pdf

Novidades e Avisos

(27-02-06) A página da disciplina está no "ar".

1 Equipa Docente e Horário

- João Saraiva http://www.di.uminho.pt/jas

Teóricas:	MCC	T1	2ª Feira - 14:00-15:00	sala CP2-102
		T2	4ª Feira - 11:00-12:00	sala CP3-103
Teórica-Prática:	MCC	TP1	2ª Feira - 15:00-16:00	sala ? ?
	MCC	TP2	4ª Feira - 09:00-10:00	sala CP3 205
Prática:	MCC	P1	3ª Feira - 09:00-11:00	sala DI 1.09
Dúvidas	MCC		3ª Feira - 14:00-18:00	

- Tiago Alves http://wiki.di.uminho.pt/wiki/bin/view/Main/TiagoAlves

Prática:	MCC	P2	3ª Feira - 11:00-13:00	sala DI 1.09
Dúvidas	MCC		? ? ?	

2 Estrutura e Funcionamento

Exposição da matéria fundamental – motivação, conceitos, definições, métodos e justificações – a nível das aulas teóricas. Resolução de exercícios de consolidação, a nível das aulas teórico-práticas, no quadro. Realização no computador, no contexto das aulas práticas, de exercícios práticos e pequenos projectos. Resolução extra aulas de dois trabalhos concretos de aplicação, recorrendo às linguagens de programação Haskell e C.

3 Objectivos

É objectivo deste curso levar os alunos a:

- Compreender os conceitos de análise e geração automática de programas.
- Aprofundar os conceitos de Gramática Independentes do Contexto e a sua utilização na especificação e no desenvolvimento de processadores de linguagens formais.
- Analisar e compreender as diferentes técnicas de parsing: "Top-Down" e "Bottom-Up"
- Analisar e compreender as tarefas de unparsing/pretty printing, análise de nomes, optimização e geração de código.
- Reforçar a aptidão dos alunos para desenvolver programas correctos e eficientes.

4 Avaliação

A Avaliação tem uma componente teórica e uma componente prática, ambas obrigatórias.

De acordo com o regulamento actualmente em vigor na UM, a **nota teórica (NT)** será obtida através da realização de **1 prova individual escrita**. Essa prova tem as instâncias a seguir indicadas (um aluno só poderá fazer melhoria na época de recurso):

- Exame, realizado na 1ª chamada da época normal, no fim do 1º semestre
- Exame, realizado na 2ª chamada da época normal, no fim do 1º semestre
- Exame, realizado na época de recurso

A **componente prática** será formada por 2 componentes

- 2 trabalhos práticos (**P1** e **P2**) a realizar durante o semestre
- Duas provas individuais escrita (**PE**) sobre os trabalhos práticos com 15 minutos de duração. Esta prova escrita será realizada imediatamente depois da entrega dos trabalhos práticos.

A nota prática (**NP**) será obtida de acordo com a seguinte fórmula:

$$\mathbf{NP} = \mathbf{P1} * \mathbf{0.4} + \mathbf{P2} * \mathbf{0.4} + \mathbf{PE} * \mathbf{0.2}$$

A nota final será determinada de acordo com a seguinte fórmula:

$$\mathbf{NF} = (\mathbf{NT} + \mathbf{NP}) / 2$$

Exige-se 8 valores como nota mínima em cada uma das partes.

5 Programa

- Processamento de Linguagens Formais
 - Conceitos e exemplos
 - Arquitectura dos Processadores de Linguagens
- Análise Léxica de Linguagens Formais
 - Conceitos e suas tarefas
 - Especificação via Expressões Regulares
 - Construção de Analisadores Léxicos baseados em Autómatos Finitos
 - Geração Automática de Analisadores Léxicos
 - * Breve Introdução ao sistema *Lex*
- Análise Sintática de Linguagens Formais
 - Conceitos e suas tarefas
 - Especificação da Estrutura de Linguagens via Gramáticas Independentes do Contexto
 - * Gramáticas/Estruturas Concretas versus Gramáticas/Estruturas Abstractas
 - Representação de Gramáticas no Paradigma Imperativo
 - * De Gramáticas Concretas para Especificações em *Yacc*
 - Breve Introdução ao sistema *Yacc*
 - * De Gramáticas Abstractas para Tipos de Dados Estruturados em *C*
 - Técnicas de Parsing
 - * Parsers Top-Down
 - * Parsers Bottom-Up
- Análise Semântica de Linguagens Formais

- Conceitos e Tarefas
- Implementação de Analisadores Semânticos: Problemas e Soluções
 - * Analisadores Semânticos baseados em Múltiplas Travessias na Estrutura Abstracta da Linguagem
 - * "Colagem" de Travessias: A solução Imperativa versus Funcional

6 Fichas Teórico-Práticas

As fichas a realizar em cada aula prática são disponibilizadas nesta secção. As fichas estarão disponíveis na sexta-feira da semana anterior à aula a que se destinam.

7 Software

O seguinte software utilizado/recomendado nesta disciplina está já disponível:

- **Sistema HaLeX:** O sistema *HaLeX* está disponível na página oficial do sistema [HaLeX.html](http://www.di.uminho.pt/~jas/Research/HaLeX/HaLeX1.1.tgz) em [HaLeX1.1.tgz](http://www.di.uminho.pt/~jas/Research/HaLeX/HaLeX1.1.tgz).
- O Sistema de Literate Programming NoWeb
 - Página do Sistema Noweb: NoWeb Homepage <http://www.eecs.harvard.edu/~nr/noweb/>
- O Gestor de Memória Boehm-Demers-Weiser
 - Página: A garbage collector for C and C++ http://www.hpl.hp.com/personal/Hans_Boehm/gc/

8 Contribuições dos Alunos

Soluções de exercícios propostos nas aulas e as melhores soluções dos trabalhos práticos serão aqui disponibilizadas.

9 Exames

Os exames em versão electrónica serão aqui disponibilizados, depois de realizados.

10 Trabalhos Práticos

O enunciado dos trabalhos práticos será aqui disponibilizado.

11 Sumários

A definir

12 Bibliografia

- **Andrew W. Appel** **Modern Compiler Implementation in C: basic techniques** *Cambridge University Press 1997*
- **Andrew W. Appel** **Modern Compiler Implementation in ML: basic techniques** *Cambridge University Press 1997*
- **Alfred V. Aho and Ravi Sethi and Jeffrey D. Ullman** **Compilers: Principles, Techniques and Tools** *Addison-Wesley 1986*
- **João Saraiva** **Language Processing (with a Functional Flavour)** *DI/UM 1ª Edicao 2000*
- **João Saraiva** **Especificação e Processamento de Linguagens** *DI/UM 1ª Edicao* [PS]_{>http://www.di.uminho.pt/~jas/mp3/esp_poc_ling.ps.gz} [HTML]_{><http://www.di.uminho.pt/~jas/Teaching/LectureNotes/esprocling.html>} *1995*
- **William M. Waite and Lynn R. Carter** **An Introduction to Compiler Construction** *Harper Collins 1993*
- **Thomas Niemman** **A Compact Guide to Lex Yacc** [PDF]_{><http://epaperpress.com/lexandyacc/download/lexyacc.pdf>}
- **M. E. Lesk and E. Schmidt** **Lex - A Lexical Analyzer Generator.** *Computing Science Technical Report No. 39, Bell Laboratories, Murray hill, New Jersey* [PDF]_{><http://epaperpress.com/lexandyacc/download/lex.pdf>} *1975*
- **Stephen C. Johnson** **Yacc: Yet Another Compiler Compiler** *Computing Science Technical Report No. 32, Bell Laboratories, Murray hill, New Jersey* [PDF]_{><http://epaperpress.com/lexandyacc/do>} *1975*
- **John Levine, Tony Mason and Doug Brown** **lex yacc** *O'Reilly Associates, Inc 1992*
- **João Saraiva** **HaLeX: A Haskell Library to Model, Manipulate and Animate Regular Languages** *ACM Workshop on Functional and Declarative Programming in Education (FDPE/PLI'02), Pittsburgh, USA,* [PS]_{><http://www.di.uminho.pt/~jas/Research/Papers/FDPE02/fdpe02.ps.gz>} [H] *2002*
- **John Hopcroft, Rajeev Motwani and Jeffrey Ullman** **Introduction to Automata Theory, Languages and Computation** *Addison-Wesley 2nd Edition Edicao* [HTML]_{><http://www-db.stan>} *2001*

- **R. Flويد and R. Beigel** **The Language of Machines: An Introduction to Computability and Formal Languages** *Computer Science Press* 1994
 – Repeated KeyFB94
- **S.P.Jones, et al** **Report on the Programming Language Haskell 98**[HTML]_{><http://www.haskell.org/onlin>} 1999
- **R. Bird** **Introduction to Functional Programming using Haskell** *Prentice Hall* 1998
- **S. Thompson** **Haskell - The Craft of Functional Programming** *Addison-Wesley* 2ª Edicao 1999
- **N. Jones, C. Gomard and P. Sestoff** **Partial Evaluation and Automatic Program Generation** *Prentice Hall*[HTML]_{><http://www.dina.kvl.dk/~sestoft/pebook/pebook.html>} 1993
- **E. Horowitz and S. Sahni** **Fundamentos de Estruturas de Dados** *Editora Campus* 1984