Teoria da Computação
(cod.2813)

Departamento de Informática
Universidade da Beira Interior

Ano lectivo 2005/2006

Esta página no formato pdf, no formato ps

1 Novidades

- Enunciado do primeiro trabalho disponível online na secção 8.
- Site do mooshak para o registo dos grupos e a resolução do primeiro trabalho: Aqui. Deverá escolher o concurso **TComp0506** e o grupo **turma0506**.
- Horário de atendimento afixado.
- Compleção dos critério de Avaliação.
- Fichas de exercícios disponíveis.
- Reposição de aulas:
  - Teórica: Quinta-Feira 13/10 das 19h às 21h
  - TP1 e TP2: Quarta-Feira 26/10 das 19h às 20h
  - P1: Quinta-Feira 27/10 das 19h às 21h

Conteúdo

1 Novidades **1**

2 Docentes **2**

3 Objectivos **2**

4 Programa **3**

5 Critérios de Avaliação **5**
5.1 Componente Ensino/Aprendizagem 5
5.2 Admissão e Avaliação por Exame 5

6 Datas Importantes **6**

7 Material Pedagógico **6**

8 Trabalho Prático **7**

9 Horário **7**

10 Atendimento **7**

11 Alunos de Engenharia Informática inscritos **8**

12 Links úteis **8**

2 Docentes

Simão Melo de Sousa (regente) - Gabinete 4.3.

3 Objectivos

Existem limites à capacidade de resolução de problemas por um computador, mesmo na hipótese "idealista" de ausência de restrições, que sejam essas o tempo (de execução) ou o espaço (memória).

Para delinear esses limites, visaremos:

1. perceber a capacidade de computação das máquinas, assim como os seus limites teóricos. Precisaremos de definir formalmente o que é e o que não é um programa, um algoritmo, ou mais genericamente o que é um tratamento efectivo;

2. perceber os conceitos que fundamentam as linguagens de programação. Precisaremos de determinar e estudar formalmente as construções que determinam a expressividade (ou capacidade de computação) das linguagens de programação assim como o comportamento dos programas.

4 Programa

Encontrará aqui o escalonamento previsto das aulas.

- Apresentação Contextual e Histórica da Disciplina
- Conceitos Preliminares
  - Conjuntos
  - Operadores sobre Conjuntos
  - Relações
  - Conjuntos Ordenados
  - Indução Bem Fundada
- Reticulados, CPOs e Pontos fixos
  - Definições e Propriedades Básicas
  - Reticulados Completos
  - Funções Monótonas e Contínuas
  - CPOs e Pontos Fixos
  - Construção de Funções como Ponto Fixos de Operadores
- Indução Estrutural
  - Conjuntos definidos por Indução, Definições Indutivas
  - Demonstração por Indução

  - Indução Estrutural e Conjuntos de Termos
  - Funções Estruturalmente Recursivas
  - Indução Estrutural, Indução Bem Fundada e Pontos Fixos
  - Caso de Estudo: Dedução Natural
- Cálculo Lambda Puro
  - Notações e Teoria Básica
  - Reduções e Igualdade
  - Propriedades
    * Confluência
    * Terminação
    * Normalização e Estratégias de Redução
  - Definibilidade Lambda
- Teoria das Funções Recursivas de Kleene
  - Funções Primitivas Recursivas
  - Funções μ-Recursivas
  - Funções Parciais
- Computabilidade e Decidibilidade o
  - Teses de Church
  - Equivalência de Modelos Computacionais
  - Problemas Decidíveis
- Para Além da Decidibilidade
  - Conjuntos Recursivos
  - Conjuntos Recursivamente Enumeráveis
  - Problemas Indecidíveis, Problemas Semi-Decidíveis
  - Reduções de Problemas

5 Critérios de Avaliação

A avaliação será realizada por **exame** e **avaliação contínua**.

Listamos a seguir as diferentes componentes da avaliação.

5.1 Componente Ensino/Aprendizagem

- A avaliação contínua mede em termos práticos a acquisição dos conceitos expostos. Como tal é baseada na realização, durante as aulas práticas da disciplnas, de **dois trabalhos** entregue à equipa docente e de uma frequência.

- O enunciado do primeiro trabalho será entregue dia 18 de Outubro e entregue (após sua resolução nas aulas práticas) nas aulas práticas da terceira semana de Novembro. O trabalho será defendido nas aulas práticas da quarta semana de Novembro.

- O enunciado do segundo trabalho será entregue dia 29 de Novembro e entregue (após sua resolução nas aulas práticas) nas aulas práticas da segunda semana de Janeiro. O trabalho será defendido nas aulas práticas da terceira semana semana do mês de Janeiro.

- A *Nota da Componente Prática* (NCP, 20 valores) é a média das notas dos dois trabalhos.

- A avaliação da acquisição de conceitos teóricos é baseado numa prova escrita, em concreto, um frequência. A frequência terá lugar na última aula teórica, Terça Feira 24 de Janeiro 2006 e resultará na atribuição da *Nota da Frequência* (NF, 20 valores) .

- Esta avaliação resultará na atribuição da Nota da Componente Ensino/Aprendizagem (NCEA, 20 valores).

  Esta nota é calculada da seguinte forma: $NCEA = \frac{NF*12+NCP*8}{20}$.

5.2 Admissão e Avaliação por Exame

- Qualquer nota $NCEA$ maior ou igual a 10 que respeita os mínimos significa aprovação a disciplina.

- Mínimos: é admitido a exame quem tiver a nota $NCEA \geq 7$. Adicionalmente, é admitido a exame mas reprovado na avaliação prática quem tiver:
  - $NCP < 7$;
  - ou $NF < 7$.
- A prova escrita do exame dará lugar a Nota da Prova Escrita (20 valores).
- A nota final (NFin) é calculada da seguinte forma:

$$NFin = if\ (NPE \geq 7)\ then\ \frac{max(NF, NPE) * 12 + NCP * 8}{20}\ else\ Reprovado$$

6 Datas Importantes

- Entrega do Enunciado do Trabalho: 18 de Outubro
- Entrega do Trabalho: P da semana do 21 de Novembro
- Defesa do trabalho: P da semana do 28 de Novembro
- Entrega do Enunciado do Trabalho: 29 de Novembro
- Entrega do Trabalho: P da semana do 10 de Janeiro
- Defesa do trabalho: P da semana do 17 de Janeiro
- Frequência: Terça Feira 24 de Janeiro 2006
- Exame Época 1 : 2006-2-6, 9:30, sala 606 (por confirmar)
- Exame Época 2 : 2006-2-20, 9:30, sala 206 (por confirmar)
- Exame Época Especial : 2006-9-4, 9:30, sala 602 (por confirmar)

7 Material Pedagógico

Apontamentos apresentados e disponibilizados nas aulas

Fichas práticas

- Ficha Exercícios Teóricos
- Ficha Programação OCaml

8 Trabalho Prático

- Primeiro Trabalho:
  - Aqui(pdf) ou Aqui(ps).
  - Regras de funcionamento:Aqui(pdf) ou Aqui(ps).
  - Site do mooshak para o registo dos grupos e a resolução do primeiro trabalho: Aqui. Deverá escolher o concurso TComp0506 e o grupo turma0506.
- Segundo Trabalho (por anunciar).

9 Horário

Tipo de aula	**Horário**	**Sala**
Teórica	Terça-Feira das 9h00 às 11h00	6.02
Teórica-Prática 1	Quarta-Feira das 8h00 às 9h00	6.18
Prática 1	Quarta-Feira das 10h00 às 12h00	6.14
Teórica-Prática 2	Quarta-Feira das 12h00 às 13h00	6.19
Prática 2	Quinta-Feira das 9h00 às 11h00	6.14
Teórica-Prática 3	Quarta-Feira das 9h00 às 10h00	6.16
Prática 3	Quinta-Feira das 11h00 às 13h00	6.14

10 Atendimento

Horário
Quarta-Feira das 14h00 às 17h00

11 Alunos de Engenharia Informática inscritos

- Alunos inscritos na disciplina
- TP1
- P1
- TP2
- P2
- Alunos inscritos (Fonte: Serviços Académicos) (todo o aluno inscrito na disciplina que não consta nesta lista deverá regularizar a sua situação junto dos Serviços Académicos)

12 Links úteis

- Ocaml
  - Site OCaml
  - Performance Ocaml:The great computer language shootout
  - OCamlMakefile: Makefile genérico para OCaml
  - Tuareg: Emacs Mode para Ocaml
  - Livro: Developing Applications With Objective Caml
  - Manual de Referência Ocaml
  - Sebentas diversas sobre Ocaml
  - Hump: Repositório de aplicações/bibliotecas/"material" OCaml
  - Documentação adicional (em françês)
- COQ
  - Página Principal do Sistema COQ
  - ProofGeneral: um Emacs Mode para COQ
  - Livro: COQ'ART: Interactive Theorem Proving and Program Development

  - Manual de Referência COQ
  - Tutorial COQ
  - Bibliotecas COQ
  - Tutorial sobre Tipos recursivos em COQ
  - CoqIDE, um IDE para COQ.

- Emacs
  - Site Emacs - XEmacs
  - Documentação Emacs - XEmacs
  - Comandos principais para Emacs 21

- Um site sobre Ciências da Computação (contem livros online)
  - Site Programming languages theory texts online

Referências

[1] Peter Aczel. An introduction to inductive definitions. In J. Barwise, editor, *Handbook of Mathematical Logic*, volume 90 of *Studies in Logic and the Foundations of Mathematics*, chapter C.7, pages 739–782. North-Holland, Amsterdam, 1977.

[2] A. Arnold and I. Guessarian. *Mathematics for Computer Science*. Prentice-Hall, 1996.

[3] F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.

[4] D. Bagley. The great computer language shootout. `http://www.bagley.org/~doug/shootout`

[5] H. P. Barendregt. *The Lambda Calculus, its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, revised edition, 1984.

[6] H.P. Barendregt. Lambda calculi with types. In S. Abramsky, Dov M. Gabbay, and T.S.E Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2, pages 117–310. Oxford University Press, New York, 1992.

[7] S. Burris and H. Sankappanavar. *A Course in Universal Algebra*. Springer Verlag, 1981. `http://www.cs.uu.nl/people/franka/ref`

[8] E. Chailloux, P. Manoury, and B. Pagano. Developing applications with objective caml. `http://caml.inria.fr/oreilly-book`, 2003.

[9] T. Coquand and P. Dybjer. Inductive definitions and type theory: an introduction. In *Foundations of Software Technology and Theoretical Computer Science*, pages 60–76, 1994. `http://www.math.chalmers.se/~peterd/papers/ESSLLI94.ps.gz`

[10] R. Cori and D. Lascar. *Mathematical Logic : A course with exercises – Part I*. Oxford University Press, 2000.

[11] R. Cori and D. Lascar. *Mathematical Logic : A course with exercises – Part II*. Oxford University Press, 2001.

[12] G. Cousineau and M. Mauny. *The functional approach to programming*. Cambridge University Press, 1998.

[13] N. J. Cutland. *Computability*. Cambridge University Press, 1980.

[14] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order: Second Edition*. Cambridge University Press, 2002.

[15] G. Dowek. *La logique*. Coll. Dominos. Flammarion, 1995.

[16] G. Dowek. Le langage mathématique et les langages de programmation. Voir, entendre, raisonner, calculer. Cité des sciences et de l'industrie, La Villette, Paris, 1997. `http://www.lix.polytechnique.fr/~dowek/Vulg/langagelangages.ps.gz`

[17] G. Dowek. Higher-order unification and matching. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume II, chapter 16, pages 1009–1062. Elsevier Science Publishers B.V., 2001.

[18] G. Dowek. Théories des types. DEA Programmation, Paris VII, 2002. http://www.lix.polytechnique.fr/~dowek/Cours/theories_des_types.ps.gz

[19] P. Fejer and D. Simovici. *Mathematical Foundations of Computer Science, Volume 1: Sets, Relations and Induction,*. Texts and Monographs in Computer Science. Springer Verlag, 1991.

[20] E. Gimenez. A tutorial on recursive types in Coq. http://coq.inria.fr/doc-eng.html

[21] J-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types*. Cambridge Tracts in Theoretical Computer Science 7. Cambridge University Press, 1988.

[22] C. Hall and J. O'Donnell. *Discrete Mathematics Using A Computer*. Springer Verlag, 2000.

[23] Chris Hankin. *Lambda Calculi: A Guide for Computer Scientists*, volume 3 of *Graduate Texts in Computer Science*. Clarendon Press, Oxford, 1994.

[24] D. R. Hofstadter. *Gödel, Escher, Bach: An Eternal Golden Braid*. Basic Books, New York, 1979.

[25] G. Huet. Constructive computation theory. Course Notes, DEA Informatique Université Bordeaux I, October 1992. http://pauillac.inria.fr/~huet/CCT/

[26] N. D. Jones. *Computability and Complexity from a Programming Perspective*. Foundations of Computing. MIT Press, Boston, London, 1 edition, 1997.

[27] J. W. Klop. Term rewriting systems. In S. Abramsky, Dov M. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2, pages 1–116. Oxford University Press, New York, 1992.

[28] W. Kneale and M. Kneale. *O Desenvolvimento da Lógica*. Fundação Calouste Gulbenkian, (1962) - 1991. Terceira Edição.

[29] J. L. Krivine. *Lambda-Calculus, Types and Models*. Ellis Horwood, 1993.

[30] H. R. Nielson and F. Nielson. *Semantics with Applications.* John Wiley & Sons, Chichester, 1993. http://www.daimi.au.dk/~bra8130/Wiley_book/wiley.html

[31] L. Pequet. Programmation fonctionelle - introduction illustrée en objective caml. http://www-rocq.inria.fr/~pecquet/pro/teach/teach.html, 2002.

[32] A. Pitts. Lecture notes on denotational semantics. University of Cambridge Computer Laboratory, 1999. http://www.cl.cam.ac.uk/Teaching/1998/DenoSema/dens.ps.gz

[33] A. Pitts. Lecture notes on computation theory. University of Cambridge Computer Laboratory, 2002. http://www.cl.cam.ac.uk/Teaching/2002/CompTheory

[34] F. Prost and G. Kahn. Checking mathematical facts with coq: Various examples of Coq proofs. http://coq.inria.fr/contribs-eng.html

[35] Didier Rémy. Using, Understanding, and Unraveling the OCaml Language. In Gilles Barthe, editor, *Applied Semantics. Advanced Lectures. LNCS 2395.*, pages 413–537. Springer Verlag, 2002. http://pauillac.inria.fr/~remy/cours/appsem/ocaml.pdf

[36] Coq Development Team. Documentation of the coq theorem prover. http://coq.inria.fr/doc-eng.html

[37] Coq Development Team. Reference manual of the Coq theorem prover. http://coq.inria.fr/doc-eng.html

[38] Coq Development Team. A tutorial of the Coq theorem prover. http://coq.inria.fr/doc-eng.html

[39] OCaml Development Team. *The Objective Caml system:Documentation and user's manual*, 2002. http://caml.inria.fr/ocaml/htmlman/index.html

[40] Isabelle Tellier. Introduction à l'informatique. http://www.grappa.univ-lille3.fr/polys/intro-info/index.html, 2001.

[41] D. van Dalen. *Logic and Structure*. Springer Verlag, Berlin, Germany, 1983.

[42] G. Winskel. *The Formal Semantics of Programming Languages: An Introduction*. Foundations of Computing series. MIT Press, Cambridge, Massachusetts, February 1993.

[43] Bertot Y and P. Castéran. Coq'art. `http://www-sop.inria.fr/lemme/Yves.Bertot/coqart.html`, 2003.

Enviar comentários e dúvidas para (retire os UUU) : desousaUUU@UUUdi.ubi.pt