

LINGUAGENS FORMAIS E AUTÓMATOS (TEORIA E FUNDAMENTOS DA COMPUTAÇÃO)

LICENCIATURAS EM INFORMÁTICA, ENSINO DE INFORMÁTICA,
E MATEMÁTICA E APLICAÇÕES
(2011/12)

PROGRAMA

1. Introdução
 - 1.1. Motivação
 - 1.2. Alfabetos
 - 1.3. Linguagens
 - 1.4. Operadores sobre linguagens
 - 1.5. A ordem lexicográfica

2. Autómatos Finitos
 - 2.1. Definição
 - 2.2. Reconhecimento de linguagens
 - 2.3. Autómatos finitos não-determinísticos (AFND's)
 - 2.4. Equivalência entre autómatos finitos e AFND's
 - 2.5. Linguagens regulares
 - 2.6. Propriedades de fecho para linguagens regulares
 - 2.7. Expressões regulares e linguagens associadas
 - 2.8. Equivalência entre as linguagens associadas a expressões regulares e autómatos finitos
 - 2.9. Lema da bombagem

3. Linguagens Livres de Contexto
 - 3.1. Gramáticas livres de contexto
 - 3.2. Linguagem livre de contexto
 - 3.3. Linguagens ambíguas
 - 3.4. Autómatos de pilha
 - 3.5. Equivalência entre as linguagens reconhecidas por autómatos de pilha e linguagens livres de contexto
 - 3.6. Lema da bombagem para linguagens livres de contexto

4. Teoria da Computabilidade
 - 4.1. Máquinas de Turing
 - 4.2. Linguagens recursivas e recursivamente enumeráveis
 - 4.3. Tese de Church-Turing
 - 4.4. Máquina de Turing não-determinística
 - 4.5. Equivalência entre máquinas de Turing e máquinas de Turing não-determinísticas
 - 4.6. Problema da Paragem e sua indecibilidade

5. Complexidade Computacional

5.1. Limites assintóticos

5.2. Definição de classes de complexidade com base no tempo de computação, e no tipo de máquina de Turing utilizada

5.3. As classes P e NP

5.4. Noção de certificado para computações não-determinísticas

5.5. Redutibilidade em tempo polinomial

5.6. Problemas NP-completos e NP-difíceis

5.7. Exemplos de problemas NP-completos

BIBLIOGRAFIA

Todos os livros que se seguem existem na biblioteca (as versões entre parêntesis, mais recentes, é que não se encontram lá)

[1] Apontamentos fornecidos pelo docente.

[2] M. R. Garey and D. S. Johnson. Computers and Intractability: a Guide to the Theory of NP-Completeness. W. H. Freeman & Co., 1979.

[3] J. E. Hopcroft, R. Motwani, and J. D. Ullman. Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, 2nd edition, 2001 (3rd edition, 2006).

[4] H. R. Lewis and C. H. Papadimitriou. Elements of the Theory of Computation. Prentice-Hall Internacional, Inc., 2nd edition, 1998.

[5] M. Sipser. Introduction to the Theory of Computation. PWS Publishing Company, 1997 (2nd edition, 2005).

AVALIAÇÃO

A avaliação terá o contributo de duas componentes:

1. Um exame final, obrigatório, cuja nota vai de 0 a 18,5 valores.
2. Dois trabalhos práticos, cada um valendo 0,75 valores, a submeter online. Mais detalhes podem ser encontrados em <http://w3.ualg.pt/~dgraca>.

A nota final da disciplina, em valores, é dada pela seguinte fórmula: Nota = nota exame + nota dos trabalhos.

Não é exigido notas mínimas e/ou frequência das aulas para efetuar o exame (embora se recomende vivamente que os alunos tirem o máximo proveito delas). Não haverá provas orais, exceto para notas superiores a 16 valores.

Situações que não estejam abrangidas por estas regras serão tratadas caso a caso.