

SCC 205 Teoria da Computação

e Linguagens Formais

Profa. Sandra Aluisio

sandra@icmc.usp.br

[http://wiki.icmc.usp.br/index.php/Scs-205\(sandra\)B](http://wiki.icmc.usp.br/index.php/Scs-205(sandra)B)

2/8/2011

Gerência do Curso

- Profa. **Sandra Aluisio** Sala: 4-208
- Atendimento: quartas das 14:00 – 17:00 h

- Sala de Aula (4005)
- Horário: Ter. 10:10/11:50 e Qui. 10:10/11:50

- Lista de Presença – Obrigatória
- 27 aulas, contando com provas
- **Presença mínima p/ aprovação e REC: 70% = 19 aulas**

Conteúdo Geral

1. Teoria das Linguagens Formais e dos Autômatos
2. Teoria da Computabilidade
3. Teoria da Complexidade

Quais são as capacidades fundamentais e limitações dos computadores?

Objetivo Geral

Dar ao aluno noção formal de algoritmo, computabilidade (decidibilidade) e dos formalismos para definição de linguagens (ou problemas), de modo a conscientizá-lo dos limites teóricos da ciência da computação.

Habilidades as serem trabalhadas

- 1) Dar subsídios para a definição de uma linguagem de programação, ou uma linguagem dedicada,
 - definir sua **gramática** e seu **reconhecedor**
 - Construir reconhecedores de programas em uma determinada linguagem de alto nível (Análise/ Front-end) - **Compiladores**.
 - apresentar as **características, propriedades e aplicações** das linguagens da hierarquia de Chomsky e de seus reconhecedores
 - Utilizaremos o pacote de ferramentas gráficas JFLAP para ajudar no aprendizado de Linguagens Formais e Autômatos (<http://www.jflap.org/>).

2) Estudar a meta-teoria da computação, ou seja, definir o que a teoria estuda e estabelecer suas limitações: **problemas indecidíveis e intratáveis**

- Definir o que a teoria estuda é definir o que é **computável**.
 - Faremos uma abordagem intuitiva de computabilidade usando a **noção de procedimento (efetivo)** e definiremos também o conceito por meio de modelos formais: o mais famoso é a **Máquina de Turing (MT)**.

– Apresentar a **hipótese de Church-Turing** (1936):

“Uma função é computável se é computável por MT” que formaliza a noção intuitiva de procedimento efetivo.

“A capacidade de computação representada pela Máquina de Turing é o limite máximo que pode ser atingido por qualquer dispositivo de computação, independentemente do curso de desenvolvimento da tecnologia” .

3) Diferenciar **procedimento** de **algoritmo** que é um procedimento que sempre pára

- Apresentar linguagens (ou problemas) **reconhecidas por MT** (recursivamente enumeráveis) e **decididas por MT** (recursivas).
- Mostrar que tipos de problemas algorítmicos são insolúveis/indecidíveis

4) Apresentar o conceito de **NP-completude** e de reduções

- nossa ferramenta principal de comparação da dificuldade entre problemas.
- Mostrar que tipos de problemas algorítmicos são intratáveis

Avaliação

- 3 provas com pesos iguais
- Datas:
 - 13/9 PROVA 1
 - 20/10 PROVA 2
 - 24/11 PROVA 3
- SUB: 1/12 – substitui a menor nota

Bibliografia

LIVROS TEXTOS

- Rosa, J.L.G. Linguagens Formais e Autômatos. Editora LTC, 2010.
- Hopcroft, Motwani & Ullman: Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, 2001. Errata do livro em: <http://www-db.stanford.edu/~ullman/ialc.html#errata>.
- Marcus Vinícius Midena Ramos, João José Neto & Ítalo Santiago Veja: Linguagens Formais - Teoria, Modelagem e Implementação, Editora: Bookman, 2009.
- Divério & Menezes. Teoria da Computação - Máquinas Universais e Computabilidade. Série Livros Didáticos 5, IF UFRGS, 2ª edição, 2000, editora SagraLuzzatto.
- Menezes, P.B. Linguagens Formais e Autômatos. Série Livros didáticos 3, IF UFRGS, 5ª edição, 2008, Editora Bookman. e-Book de Linguagens Formais & Autômatos.
- Sipser, M. Introduction to the Theory of Computation. PWS, 1997. (2ª edição). Errata do livro em: <http://www-math.mit.edu/~sipser/itoc-errs1.2.html>
- Harel, D. Algorithmics - The Spirit of Computing. Addison-Wesley Publishing Company, 1992 (2ª edição). (Existem 3 edições similares do livro na Biblioteca do ICMC: a primeira, a sua versão reduzida e a segunda edição que traz exercícios).

BIBLIOGRAFIA COMPLEMENTAR

- Christos H. Papadimitriou, *Computational Complexity*, Addison-Wesley Publishing Company, 1995. (Biblioteca Central).
- Cormen, T.H. Leiserson, C.E. and Rivest, R.L. *Introduction to Algorithms*. The Mit Press. 1990. (1ª edição). Capítulos 1, 2, 3, 4, 36, 37.
- Savage, J.E. *Models of Computation - Exploring the Power of Computing*. Addison-Wesley, 1998.
- Toscani & Veloso. *Complexidade de Algoritmos*. Série Livros Didáticos 13, IF UFRGS, 1ª edição, 2001, editora SagraLuzzatto.
- Garey & Johnson. *Computers and Intractability - a guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York, 1979.
- Jones, N. D. *Computability and complexity - from a Programming Perspective*. The Mit Press, 1997.
- Drobot, V. *Formal Languages and Automata Theory*. Computer Science Press, 1989.
- Sedgewick, R. and Flajolet, P. *An Introduction to the Analysis of algorithms*. Addison-Wesley P. Company, 1996.
- Sudkamp, T. A. *Languages and Machines - An Introduction to the Theory of Computer Science*, 2a edition. Addison-Wesley, 1998.
- Kozen, D.C. *Automata and Computability*. Springer-Verlag, 1997.

Programa detalhado do curso

1. Gramáticas: definições e hierarquia de Chomsky com exemplos
2. A sentença vazia e as GSC, GLC e GR
3. Sistemas de Estados Finitos e AF Determinísticos
4. AF Não-determinísticos e a Equivalência entre AFDN e AFD
5. AF com λ -movimentos e a Equivalência entre GR e AF

6. Operações Fechadas sobre LR's
7. Expressões Regulares (ER)
8. Conversão de AF para ER no JFLAP e a Equivalência entre AF e GR
9. O Utilitário Lex e as ER estendidas
10. Lema do Bombeamento: Linguagens regulares e não-regulares
- 11 Minimização de AFD

12. Autômatos com Pilha: ACPDet X ACPND
13. Notação gráfica para ACP
14. Forma Normal de Chomsky (FNC), Forma Normal de Greibach (FNG) e a Equivalência entre GLC e ACPND: os 2 teoremas
15. Árvores de Derivação para GLC e a Ambigüidade nas GLC
16. Precedência, Prioridade e Associatividade de operadores e sua aplicação na gramática de expressões das Linguagens de Programação
17. Notação EBNF (BNF estendida) e a sua aplicação em geradores de parsers descendentes recursivos e a Notação usada com o YACC (gerador de parsers ascendentes)

18. Máquinas de Turing Determinísticas
19. Máquinas de Turing com Fita Limitada (ou Autômato Limitado Linearmente – ALL) e sua relação com Linguagens Sensíveis ao Contexto
20. Usos da MT para calcular funções e para processar problemas de decisão (procedimento)
21. Máquinas de Turing com Múltiplas Fitas
22. Máquinas de Turing Não-determinísticas
23. Procedimentos e Algoritmos e a Tese de Church-Turing

24. Máquina de Turing Universal (MTU)
25. O Problema da Parada e as Provas da Indecidibilidade do Problema da Parada (contradição, diagonalização)
26. Teorema de Rice --- a indecidibilidade do Problema da Parada não é um fenômeno único
27. Outra técnica para provar indecidibilidade:
Princípio/Método da Redução
28. Outros problemas indecidíveis e parcialmente decidíveis

29. Análise de Algoritmos e o comportamento Assintótico de Funções
30. Problemas de Decisão e as classes P, NP, co-NP e suas relações
31. Algoritmos Não-determinísticos
32. Completude e a Redução Polinomial
33. Estrutura de Prova de alguns problemas NP-completos