
Descrição da Disciplina

Esta disciplina abordará modelos computacionais (finitos) usados para descrever linguagens (de computação) infinitas. Um modelo computacional finito é, essencialmente, um programa. A linguagem descrita pelo modelo são os resultados obtidos ao “executarmos” o programa, respeitada a “arquitetura” do modelo. O foco será sempre nas (classes de) linguagens. O modelo será encarado como um veículo para representar a linguagem.

As perguntas serão do gênero: “quais linguagens posso especificar usando este modelo?”; “como construir um modelo desta arquitetura para representar esta linguagem?”; “quão simples pode ser um modelo desta classe para representar esta linguagem?”; “que propriedades tem a classe de linguagens representável por este modelo?”; etc.

Os modelos que vamos estudar se dividem em dois grandes grupos: gramaticais (veremos o que vem a ser isto) e autômatos (também veremos o que são). Dentro de cada um destes grupos, vamos estudar vários modelos específicos.

Estes modelos encontram muitas aplicações práticas, em áreas bem diversas em Ciências da Computação. Por exemplo, modelos regulares (veremos o que são) são muito usados em analisadores léxicos (fazem parte de todo compilador) e em editores de texto que usamos todo dia. Modelos livre de contexto (vamos encontrá-los) formam uma parte central de todo compilador. Na verdade, toda a sintaxe de linguagens de programação é, usualmente, expressa em um destes modelos livre de contexto. Modelos regulares são muito usados em engenharia, para lidar com circuitos (digitais) sequenciais. Variações deste modelo aparecem na verificação de propriedades de programas reativos (tipo de programas que interagem com um ambiente externo, como sistemas operacionais) e também em programas de tempo real. Outras classes de modelos, mais poderosos, são extensivamente usados na análise de complexidade de problemas. A lista é longa.

Professor: **Arnaldo Moura**, IC Sala 41, e-mail: arnaldo@dcc.unicamp.br

Ementa Detalhada da Disciplina

1. Revisão de conceitos básicos
 - (a) indução
 - (b) relações
 - (c) fecho de relações
2. Alfabetos e linguagens
3. Linguagens Regulares
 - (a) gramáticas lineares à direita e à esquerda
 - (b) autômatos finitos
 - (c) não determinismo
 - (d) minimização de autômatos finitos
 - (e) equivalência de modelos
 - (f) teorema da iteração
 - (g) propriedades de linguagens regulares
4. Linguagens livre de contexto
 - (a) gramáticas livres de contexto
 - (b) derivações e árvores de derivação
 - (c) ambigüidade
 - (d) formas normais para gramáticas livres de contexto
 - (e) autômatos de pilha
 - (f) equivalência de modelos
 - (g) teorema da iteração
 - (h) propriedades de linguagens livre de contexto
5. Linguagens recursivas e linguagens recursivamente enumeráveis
 - (a) máquinas de Turing
 - (b) restrições e extensões para máquinas de Turing
 - (c) construção de máquinas de Turing
 - (d) linguagens recursivas
 - (e) linguagens recursivamente enumeráveis
 - (f) máquina universal

- (g) gramáticas sensíveis ao contexto
 - (h) gramáticas irrestritas
 - (i) hierarquia de Chomsky
 - (j) equivalência e modelos
6. Computabilidade e Decidibilidade
- (a) problemas de decisão
 - (b) problemas decidíveis, parcialmente decidíveis e indecidíveis
 - (c) problema da parada em máquinas de Turing
 - (d) reduções
 - (e) problema da correspondência de Post
 - (f) problemas indecidíveis e linguagens livre de contexto
7. [Opcional] Complexidade computacional
- (a) complexidade de tempo
 - (b) complexidade de espaço
 - (c) problemas intratáveis

Bibliografia Comentada

1. MATERIAL ESPECÍFICO DA DISCIPLINA

- (a) Notas de Aula: a critério de cada um. Eventualmente, poderão ser distribuídos textos sobre tópicos específicos.
- (b) D. Kelly, *Automata and Formal Languages*, Prentice-Hall, 1995.
Cobre todos os tópicos que discutiremos. Contém exercícios para cada seção, além de exercícios comuns por capítulos. Nível introdutório, fácil de entender. Bastante figuras e exemplos. Alguns tópicos serão tratados em mais profundidade nesta disciplina.
- (c) M. Sipser, *Introduction to the Theory of Computation*, PWS, 1997.
Um ótimo texto. Contém um capítulo inicial (Cap. 0) bastante completo em relação aos pré-requisitos mínimos necessários para o entendimento da disciplina. Texto com excelente apresentação gráfica. Vários exercícios por capítulo, separados em dois grupos, por grau de dificuldade. Adota estratégia interessante quanto às provas formais que desenvolve, oferecendo inicialmente uma explicação informal sobre a *idéia* das provas e, em seguida, passando à demonstração.
- (d) J.E. Hopcroft e J.D. Ullman, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, 1979.
Um texto clássico e muito bom. Contém exercícios por capítulos. Nível intermediário e cobre satisfatoriamente todos os tópicos que vamos estudar (Capítulos 0 a 9). Contém outros capítulos (Capítulos 10 a 14), com informações interessantes sobre tópicos adicionais válidos, inclusive, para outras disciplinas (como MC448, MC538 e MO417).
- (e) N. J. Vieira, *Introdução aos Fundamentos da Computação: Linguagens e Máquinas*, Thomson, 2006.
Um texto em português, bastante bom e com vários exercícios. Contém exercícios por capítulos. Cobre satisfatoriamente a maioria todos os tópicos que vamos estudar (caps. 1 a 5).
- (f) H.R. Lewis e C.H. Papadimitriou, *Elements of the Theory of Computation*, Prentice-Hall, 1998.
Um livro clássico e muito bom. Exercícios são subdivididos por seções. Há também uma tradução para o português pela Bookman; neste caso use apenas da segunda edição (revisada) em diante (caps. 1 a 5).
- (g) R.Greenlaw, H.J.Hoover *Fundamentals of the Theory of Computation*, Morgan Kaufmann, 1998.
Texto de dificuldade intermediária. Bem didático, com muitos exercícios separados por seção. Contém dois capítulos iniciais com material introdutório difícil de encontrar em outros textos no mesmo nível de detalhe. Bons capítulos para quem precisa de um reforço nessa parte inicial. Cobre quase todos os tópicos que estudaremos; não cobre a parte final, que trata de computabilidade e indecidibilidade.
- (h) M.A. Harrison, *Introduction to Formal Language Theory*, Addison-Wesley, 1978.
Texto mais avançado. Tratamento matemático mais rigoroso e extenso. Contém exercícios por seção. Cobre tópicos além do que estudaremos, especialmente na área importante de linguagens livre de contexto determinísticas, muito usadas na especificação de linguagens de programação, na prática.

- (i) R.W. Floyd e R. Beigel, *The Language of Machines*, W. H. Freeman, 1994.
Um texto bem escrito. A notação é um tanto não ortodoxa, uma vez que o livro tenta unificar vários tratamentos sob a mesma notação. Contém bons exercícios. Bom como fonte de referência.
- (j) M. Minsky, *Computation: Finite and Infinite Machines*, Prentice-Hall, 1967.
Um texto mais antigo, porém muito bom e legível. Abordagem um tanto diferenciada, enfocando mais questões de computabilidade. Contém exercícios por seção, inclusive algumas soluções.
- (k) I. Simon, *Linguagens Formais e Autômatos*, Segunda Escola de Computação. Campinas, 1982.
Em português. Abordagem e notação próprias. Bastante consiso e denso. Mais raro e difícil de encontrar.

2. TEXTOS DE APOIO E MATERIAL DE REVISÃO

- (a) F. Preparata e R. Yeh, *Introduction to Discrete Structures*, Addison-Wesley, 1973.
Ótimo texto introdutório aos conceitos de conjuntos, funções, relações e indução matemática (Capítulos 0 e 1). Bem explicativo e claro. Muitos exercícios.
- (b) A. Gill, *Applied Algebra for the Computer Sciences*, Prentice-Hall, 1976. Capítulos 1, 2 e 3. O livro é muito bom, cheio de exercícios, separados por cada seção. A apresentação é muito boa. Há vários exemplos discutidos no corpo do texto. Eventualmente, o Capítulo 8 também poderá ser útil no decorrer da disciplina.
- (c) J. Loeckx e R. Sieber, *The Foundations of Program Verification*, 2ª edição, Wiley-Teubner, 1984.
Ótimo texto. Embora esteja voltado para a área de semântica e verificação de programas (no que é muito bom), o Capítulo 1 contém uma excelente introdução aos pré-requisitos básicos que vamos precisar conhecer, especialmente indução matemática. Contém exercícios por capítulo.
- (d) U. Manber, *Introduction to Algorithms*, Addison-Wesley, 1989.
Texto específico sobre algoritmos. O Capítulo 1 contém boa introdução à indução matemática. Porém com forte viés para a área de algoritmos. Muitos exercícios, alguns com sugestões de solução. Certamente será útil também em outras disciplinas que versem sobre algoritmos.
- (e) R. Graham, D. Knuth e O. Patashnik, *Concrete Mathematics*, 2ª edição, Addison-Wesley, 1994.
Excelente texto, com muitos exercícios, inclusive com soluções. Cobre vários tópicos e técnicas na área de matemática discreta, importantes para Ciências da Computação, em geral.

Organização e Avaliação

1. HORÁRIO PREFERENCIAL DE ATENDIMENTO: após as aulas.
Se necessitar de um horário diferenciado, comunique-se via e-mail.
2. AVALIAÇÃO: A avaliação da disciplina será baseada nos seguintes itens:
 - (a) Listas de Exercícios: serão distribuídas listas de exercícios periodicamente, dependendo do desenvolvimento dos tópicos em aula. Questões de provas poderão — deverão — ser baseadas em problemas que constam das listas. Além disto, as listas servem para aquilatar seu entendimento dos tópicos discutidos em sala de aula. Os problemas propostos nas listas podem ser discutidos livremente. As listas não serão recolhidas, mas aconselhamos, *fortemente*, que cada aluno se empenhe em resolver os problemas propostos. Problemas selecionados poderão ter sua solução comentada ou distribuída.
 - (b) Provas: Vamos realizar três provas (de 2 horas) nos dias 17.04, 17.05 e 19.06.
 - (c) Exame (alunos de graduação): O exame final será realizado no dia 10.07, na mesma sala e hora da aula.

Sua média no semestre será computada como $MS = (1.5P_1 + 2.0P_2 + 2.5P_3)/6.0$, onde P_i , $i = 1, 2, 3$, é a nota das provas.
3. ALUNOS DE GRADUAÇÃO: Se você terminar o semestre com $MS \geq 5.0$, terá SE APROVADO na disciplina e sua nota final será $F = MS$. Caso contrário, você deverá prestar o exame final (E) e sua nota final será computada como $F = (MS + E)/2.0$; você terá SE APROVADO na disciplina se conseguir $F \geq 5.0$, caso contrário terá SE REPROVADO na disciplina.
4. ALUNOS DE PÓS-GRADUAÇÃO: Seu conceito final será obtido a partir de MS , *aproximadamente* de acordo com o seguinte critério: se $MS \geq 8.0$, conceito A; se $5.5 \leq MS < 8.0$, conceito B; se $3.0 \leq MS < 5.5$, conceito C; se $MS < 3.0$, conceito D.