

Prefácio

O presente livro, obra cuidadosamente produzida como fruto de diversos anos de trabalho dos autores, corresponde a uma compilação aprimorada do material por eles utilizado e extensivamente testado em disciplinas de graduação e de pós-graduação ministradas na Escola Politécnica da Universidade de São Paulo e também em outras instituições de ensino superior, de Engenharia Elétrica com ênfase em Sistemas Digitais, de Engenharia de Computação e de Bacharelado em Ciência da Computação.

O material aqui apresentado tem como finalidade principal servir como texto fundamental de estudo sobre a teoria das linguagens formais e das correspondentes máquinas computacionais tradicionais.

Seu público-alvo é, portanto, o aluno de ciências exatas, cursando as disciplinas de fundamentos da computação que constituem a estrutura teórico-conceitual de sua particular área de interesse.

Não é difícil notar em alguns cursos de computação uma tendência relativamente recente, manifesta através de perigosos desvios curriculares que, em nome de questionamentos quanto à utilidade imediata de temas ligados às bases da área, reduzem drasticamente a atenção do curso em relação a matérias de formação centradas nos fundamentos matemáticos da computação.

Isso acarretaria para o aluno uma formação precária e deformada, pois, carente de alicerces conceituais da área em que atuar no dia-a-dia, o aluno se vê cercado por uma infinidade de métodos e técnicas, aparentemente desconexos, a serem mecanicamente aplicados sem que se dê conta da estreita relação que guardariam entre si.

O conhecimento e o domínio de disciplinas teóricas, embora possam ser considerados por muitas correntes como desnecessários, cada vez mais se confirmam como indispensáveis para a formação de profissionais competentes.

Com a intenção de dar aos professores melhores condições para que não se perca essa riqueza proporcionada pelos fundamentos computacionais na formação dos alunos, optamos por redigir este livro de tal forma que, sem perda de rigor ou de conteúdo, mas com substancial redução da dificuldade formal, seja possível apresentar a matéria do seu conteúdo de uma forma muito mais leve e suave do que se encontra na maioria das publicações congêneres disponíveis no mercado.

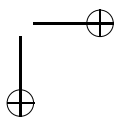
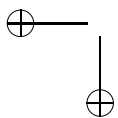
Para tanto, no presente livro procurou-se oferecer ao leitor diversas características muito desejáveis e pouco encontradas na literatura técnica atualmente comercializada em língua vernácula.

Assim, uma combinação muito cuidadosa de teoria e prática permeia todo o material, eliminando a aridez encontrada na maioria dos textos similares, porém sem cair no extremo de valorizar excessivamente a parte prática em detrimento da teórica.

Como exemplo, a apresentação dos teoremas neste livro colocará sua ênfase principal nas hipóteses em que eles se aplicam e nos resultados deles auferidos, sendo a natureza das correspondentes demonstrações, sempre que conveniente, preferencialmente algorítmica em vez de algébrica, para facilitar o entendimento e a assimilação dos aspectos práticos da aplicação dos teoremas em questão.

O emprego de demonstrações formais, excessivamente detalhistas, longas e abstratas foi, portanto, evitado, sendo elas substituídas pelo uso de provas elaboradas no estilo construtivo, com a finalidade de, em muitos casos, utilizar o método empregado nessas provas também como roteiro para a construção de programas que implementam, no computador, o mesmo processo utilizado para provar os teoremas.

Com a finalidade de reduzir a aura de mistério historicamente cultivada em torno de diversos assuntos que, embora importantes, têm sido sistematicamente evitados na maioria das obras disponíveis no mercado, foi incluída neste livro uma grande quantidade de informações correspondentes, de escla-



recimentos e de detalhes, nem sempre triviais, da teoria que costumam ser sumariamente omitidos ou deixados para o leitor resolver.

Um grande número de exemplos e de exercícios resolvidos complementa bem a teoria, auxiliando o leitor em seu aprendizado, sem procurar ocupar a posição dos assuntos conceituais dos quais derivam.

Um significativo diferencial deste livro em relação a seus congêneres é a presença de uma importante componente experimental, que, utilizando uma linguagem de programação para implementar a teoria na forma de programas executáveis, torna prático exercitar no computador, como em um laboratório, os principais conceitos teóricos apresentados.

Freqüentemente textos sobre o assunto tratado neste livro costumam ser apresentados com um enfoque preponderantemente algébrico, que demanda uma grande capacidade de abstração e de raciocínio lógico, a qual, muitas vezes, torna-se obstáculo adicional para uma sólida compreensão da matéria. Nesse sentido, a componente experimental apresenta diversas vantagens e contribui de várias formas para uma melhor fixação do assunto.

Entre os autores que seguem essa linha, é comum fazer-se uso de experimentos apoiados em simuladores gráficos que permitem ao aluno construir e observar o comportamento dos dispositivos suportados pelas ferramentas em questão. Um exemplo típico é a ferramenta JFLAP (<http://www.jflap.org/>).

A componente experimental adotada nesta obra é diferente e baseia-se na modelagem dos dispositivos estudados na teoria e na posterior implementação dos mesmos através de programas de computador.

Empregou-se para isso a linguagem de programação Ruby ([1]), caracterizada por ser dinâmica e interpretada, com suporte à orientação a objetos. Essa opção levou em conta tais características da linguagem, e o conforto que delas advém devido à interatividade e facilidade de manipulação que proporciona. A linguagem de modelagem adotada foi a UML (*Unified Modeling Language*, [2]).

As vantagens decorrentes do emprego das técnicas de modelagem e de programação no desenvolvimento da componente experimental do livro são: (i) a aproximação da teoria da computação com a engenharia de software estimula a curiosidade do aluno, motiva-o a realizar os experimentos propostos no texto e proporciona-lhe uma perspectiva familiar para a compreensão dos conceitos; (ii) a disponibilização de modelos e dos correspondentes códigos-fonte faz com que o aluno seja capaz, inicialmente, de compreender as decisões de projeto e, num segundo momento, de interferir nesse ambiente, seja para modificar ou para estender a sua funcionalidade. Finalmente, ele se torna apto a projetar e implementar novos experimentos, usando para isso as técnicas discutidas no livro.

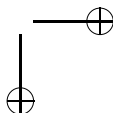
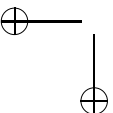
Não houve em nenhum momento a intenção de apresentar nesta publicação todos os recursos da linguagem Ruby, e por essa razão foram incluídos, na medida da necessidade para a construção do código-fonte, estritamente aqueles que se mostraram necessários à realização dos modelos implementados.

Note-se que, embora tenha havido uma permanente preocupação de manter a maior aderência possível entre a teoria e o código-fonte correspondente em Ruby, nem sempre essa prática conduziria a implementações que apresentassem a clareza desejada quanto aos conceitos teóricos subjacentes.

Como conseqüência dessa busca de clareza para o código-fonte quanto ao entendimento dos aspectos teóricos que representam, surgiram diferenças pragmáticas que poderão ser observadas entre os modelos clássicos e os correspondentes modelos de implementação adotados.

Isso, porém, não provocou prejuízos aos conceitos, mas apresentou a vantagem de manter, nas implementações dos diferentes experimentos, uma razoável uniformidade nas abstrações empregadas nos modelos de implementação.

Completando, uma vasta gama de comentários explicativos detalhados interpreta a maioria dos pontos teóricos em que o leitor típico costuma ter suas maiores dificuldades de compreensão.



O estudo de linguagens formais e autômatos compreende os aspectos teóricos ligados à classificação e à formalização das linguagens estruturadas em frases — sejam elas naturais, de programação ou outras —, bem como a análise de suas propriedades e de seus respectivos modelos de implementação.

Linguagens formais e autômatos constituem a base teórica da computação, com influência direta sobre várias outras disciplinas, em particular as de projeto de linguagens de programação, construção de compiladores e reconhecimento de linguagens naturais, servindo ainda como base para matérias como projeto de sistemas digitais, circuitos lógicos, fundamentos de sistemas de comunicação etc.

Este livro está organizado em quatro partes: na primeira (Capítulos 1 e 2, exceto as Seções 1.8 e 2.7) são apresentados os fundamentos matemáticos e os conceitos básicos necessários ao estudo de linguagens formais e autômatos. A segunda parte (Capítulos 3, 4, 5, 6 e 7, exceto as Seções 3.12, 4.15, 4.16, 5.10, 5.11, 6.8 e 7.11) compreende a teoria clássica de linguagens formais e autômatos, sendo desenvolvida em uma escala de complexidade crescente das várias classes de linguagens consideradas e abrangendo, para cada uma delas, tópicos como gramáticas, reconhecedores e propriedades. A terceira parte (Seções 4.15 e 5.10) discute modelos teóricos alternativos de implementação para linguagens formais, e seu material constitui resultado de pesquisa original. A quarta parte (Seções 1.8, 2.7, 3.12, 4.16, 5.11, 6.8 e 7.11) apresenta e discute questões ligadas à modelagem e à implementação das várias classes de linguagens formais consideradas, através do uso das linguagens UML e Ruby.

O livro foi elaborado para ser usado como texto de referência em disciplinas de linguagens formais e autômatos (e afins) em cursos de graduação e pós-graduação na área de computação. Esse foco compreende a primeira e segunda partes do livro, sendo que o Capítulo 1 poderá ser omitido caso o estudante já tenha tido contato com os assuntos da matemática discreta.

A terceira parte auxilia a ampliar as fronteiras da teoria clássica através da inclusão de dispositivos formais adicionais, que apresentem vantagens, quer do ponto de vista de formalização quer do de desempenho, em relação àqueles mais clássicos, usualmente estudados na teoria.

Para aqueles que já tenham tido um primeiro contato com a teoria clássica, a parte quatro do livro pode ser aproveitada como roteiro para o desenvolvimento de atividades experimentais de modelagem e implementação das principais abstrações estudadas no livro. Essa aproximação entre as áreas de teoria da computação e da engenharia de software é intencional, propiciando a exploração de novas e interessantes possibilidades didáticas, em ambas as áreas. Conforme o tempo disponível, os dois usos podem ser combinados em uma mesma disciplina.

Todas as linguagens (formais) podem ser vistas como conjuntos — finitos ou infinitos — de sentenças de comprimento finito. A formalização das linguagens consiste na utilização de métodos matemáticos, portanto rigorosos, para a obtenção de definições, completas e precisas, da estrutura das sentenças. Os conceitos e as ferramentas matemáticas básicas utilizadas no desenvolvimento desses assuntos são apresentados no Capítulo 1.

As linguagens finitas, ou seja, aquelas formadas por um conjunto finito de sentenças, podem ser formalizadas através da simples enumeração das sentenças que as compõem. Se a quantidade de sentenças for pequena, o problema torna-se trivial.

Dificuldades práticas surgem, no entanto, no caso da maioria das linguagens de maior interesse, pois elas costumam ser formadas por uma quantidade infinita de sentenças, ou, ao menos, por uma quantidade finita porém exageradamente grande de sentenças. Em ambos os casos, a definição de tais linguagens através de enumerações torna-se inviável, e isso motiva a busca de representações compactas para a sua formalização.

Definições finitas para linguagens infinitas podem ser encontradas em duas classes fundamentais de dispositivos formais: os de geração de sentenças, conhecidos como gramáticas, e os de aceitação de sentenças, conhecidos como reconhecedores. O Capítulo 2 apresenta e formaliza os conceitos básicos gerais de linguagens, gramáticas e reconhecedores.

Grande parte do estudo de linguagens formais e autômatos adota para as linguagens uma classificação que leva em conta o seu grau de complexidade. Tal classificação foi elaborada por Noam Chomsky

no final da década de 1950. Conhecida como Hierarquia de Chomsky, ela é apresentada também no Capítulo 2.

Os capítulos seguintes são dedicados ao estudo das quatro diferentes classes de linguagens definidas na Hierarquia de Chomsky: as linguagens regulares, analisadas no Capítulo 3, as linguagens livres de contexto, no Capítulo 4, as sensíveis ao contexto, no Capítulo 5, e as recursivamente enumeráveis, no Capítulo 7. A classe das linguagens recursivas, originalmente não incluída na Hierarquia de Chomsky, porém relevante para uma compreensão mais abrangente do assunto, é analisada no Capítulo 6.

Para cada uma dessas classes de linguagens individualmente consideradas, serão estudados aspectos de conceituação e de formalização gramatical, suas propriedades mais importantes, e questões ligadas à correspondente classe de reconhecedores.

De uma forma geral, serão discutidos, em cada um desses capítulos, as características estruturais típicas da classe de linguagens considerada, a sua formalização gramatical e as respectivas formas normais mais importantes (se aplicáveis), o dispositivo de reconhecimento associado e a equivalência das metalinguagens (cognitivas e generativas) apresentadas. Na seqüência, discutir-se-á a relação da presente classe de linguagens com a classe de linguagens imediatamente mais restrita (conforme o capítulo imediatamente anterior), e a existência de linguagens que não pertencem à classe considerada, servindo dessa forma como introdução para a discussão da classe de linguagens imediatamente mais abrangente (conforme o capítulo imediatamente posterior). Para encerrar, serão estudadas as principais propriedades da presente classe de linguagens, englobando questões decidíveis e propriedades de fechamento. No Capítulo 4, sobre linguagens livres de contexto, serão ainda caracterizadas as subclasses das linguagens livres de contexto determinísticas e das linguagens livres de contexto não-ambíguas, particularmente relevantes no campo da especificação e da implementação de processadores para linguagens de programação.

Essa abordagem confere ao texto um caráter didático importante, na medida em que permite que o leitor identifique e assimile, passo a passo, e em uma seqüência de complexidade crescente, as classes de linguagens mais importantes e, especialmente, as relações que existem entre elas.

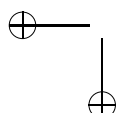
As Seções 4.15 e 5.10 apresentam, respectivamente, duas interessantes variantes do modelo de implementação para linguagens livres de contexto (Seção 4.15) e linguagens sensíveis ao contexto (Seção 5.10). Trata-se do autômato de pilha estruturado, um formalismo muito conveniente para a representação e implementação de linguagens regulares e livres de contexto, e do autômato adaptativo, que estende o anterior, conferindo-lhe a capacidade de representar e formalizar dependências de contexto, tornando-o assim adequado para a representação de linguagens de alta complexidade, inclusive as recursivamente enumeráveis.

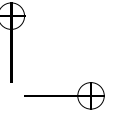
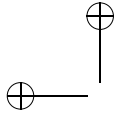
O Capítulo 8 oferece, finalmente, uma retrospectiva panorâmica de todo o material apresentado neste livro, sob a ótica da teoria da computação, incluindo uma breve introdução ao assunto complexidade.

Os principais conceitos de cada capítulo são ilustrados, de forma prática, através de exemplos modelados na linguagem UML e implementados na linguagem Ruby (Seções 1.8, 2.7, 3.12, 4.16, 5.11, 6.8 e 7.11). Esses exemplos permitem que o leitor vivencie, por meio de simulações no ambiente computacional, os conceitos e os modelos matemáticos apresentados no texto de forma abstrata, contribuindo assim para uma melhor assimilação do material teórico.

Ao final de cada capítulo, um conjunto de exercícios propostos auxilia o leitor a fixar os conceitos e a materializar sua aplicação prática, enriquecendo assim a formação obtida pela leitura.

Temos certeza de que esta obra irá contribuir significativamente para a desmistificação de muitos tópicos da teoria da computação tradicionalmente considerados complexos ou até mesmo inatingíveis, dando-lhes uma interpretação prática e laboratorial, portanto computacionalmente executável, mostrando assim não apenas o significado de cada um desses tópicos, mas apontando também o caminho a seguir para utilizá-los no dia-a-dia, sem receio e com conhecimento de causa.





Desejamos aos professores que tirem o melhor proveito deste nosso trabalho em suas aulas, e pedimos que nos ajudem a aprimorá-lo informando-nos das inevitáveis imprecisões ou incorreções que vierem a encontrar na obra, e sugerindo melhorias para as próximas edições.

Aos alunos e profissionais que utilizarem nosso livro em seus estudos, solicitamos que nos indiquem as dificuldades encontradas no uso do mesmo, para que possamos cada vez mais aperfeiçoar a didática e manter disponível um texto útil e eficaz para o aprendizado de tão importante matéria.

