

Ensino de linguagens formais e autômatos em cursos superiores de computação

Marcus Vinícius Midená Ramos

Resumo

Linguagens formais e autômatos é uma disciplina fundamental dos cursos superiores da área de computação, especialmente daqueles que apresentam ênfase na formação científica do aluno, como é o caso dos cursos de bacharelado em Ciência da Computação e de vários cursos de Engenharia de Computação. Ela faz parte do núcleo denominado “Fundamentos da Computação” (conforme o currículo de referência da Sociedade Brasileira de Computação [1]), e contempla tópicos que, normalmente, são considerados, pelos alunos, excessivamente áridos, abstratos, complexos e desvinculados da sua futura realidade profissional, o que contribui para reduzir o interesse e a motivação pelo seu aprendizado. O objetivo do presente artigo é refletir sobre os conteúdos programáticos usualmente adotados para o ensino do seu conteúdo, oferecendo alternativas que possam contribuir para uma melhor assimilação dos tópicos da disciplina. Inicialmente é feita uma análise da enfoque comumente adotado para a construção e o desenvolvimento do referido conteúdo. A seguir, apresentam-se propostas que possam contribuir para uma maior motivação dos alunos para o estudo da disciplina, assim como para uma aprendizagem mais eficiente.

Palavras-chave: *linguagens formais, autômatos, ensino, aprendizagem.*

Abstract

Formal languages and automata theory is a fundamental discipline in undergraduate computer programs, specially those with emphasis in the scientific education of the student, such as Computer Science and Computer Engineering . The discipline belongs to the kernel known as “Foundations of Computing” (according to the Brazilian Computer Society [1]), and contains a body of knowledge which is normally considered, by its students, as highly abstract, complex and unrelated to their future professional activities, thus reducing their interest and motivation for its study. The objective of the present article is to reflect upon common practices adopted in the development of its contents, presenting some alternatives that may contribute for a better absorption of the discipline. It starts with an analysis of the most usual approaches adopted, and then presents proposals that might contribute for a higher degree of motivation of the students, as well as for a more efficient learning process.

Keywords: *formal languages, automata, teaching, learning.*

UNIVASF — Universidade Federal do Vale
do São Francisco

Colegiado do Curso de Engenharia da
Computação

Avenida Antonio Carlos Magalhães,
510 — Santo Antônio
Juazeiro — BA
CEP 48902-300

marcus.ramos@univasf.edu.br

1 Introdução

Linguagens formais e autômatos constitui matéria considerada “clássica” na área da computação, uma vez que os seus principais resultados foram consolidados nas décadas de 1960 e 1970, portanto há mais de 30 anos. Como resultado, existe um enfoque também “clássico” utilizado na produção de conteúdos programáticos e de estruturas curriculares voltados para o ensino dessa matéria. Textos didáticos (tais como livros, apostilas e apresentações) costumam adotar enfoques igualmente “clássicos”, com pouca diferenciação e inovação. Tais enfoques deixam, por vezes, de explorar caminhos que podem contribuir para aprimorar a qualidade da aprendizagem dos alunos. O objetivo do presente artigo é analisar o enfoque clássico, apresentando e analisando algumas sugestões para a construção de conteúdos programáticos (e, consequentemente, também, de textos didáticos) que sejam voltados para um ensino de linguagens formais e autômatos mais eficaz e mais estimulador do aprendizado.

2 Formação teórica

O área da computação tornou-se, em pouco mais de 60 anos, um verdadeiro universo. O profissional moderno pode atuar em sub-áreas muito diversificadas, as quais requerem conhecimentos específicos e aptidões também bastante variadas.

Em função de toda essa diversidade, e também da complexidade a ela associada, os cursos superiores de computação se deparam com o dilema de procurar, com grandes esforços, acompanhar as inovações e as tendências tecnológicas da área, incorporando-as de alguma forma aos seus currículos, ou, então, de oferecer aos seus alunos uma sólida formação acadêmica, sobre a qual o futuro profissional possa construir uma também sólida carreira, conforme a sub-área em que pretenda atuar.

Se, por um lado, os cursos que possuem uma ênfase maior nos aspectos tecnológicos da computação favorecem a empregabilidade do aluno recém-formado, por outro lado esses mesmos alunos se ressentem, à medida que o tempo passa, da falta de uma formação teórica mais completa que lhes permita se renovarem e se manterem competitivos no mercado de trabalho por mais tempo.

Por esse motivo, o estudo de matérias teóricas, como é o caso de Linguagens Formais e Autômatos, não pode ter a sua importância relativizada na estrutura curricular de cursos que pretendam formar profissionais preparados para enfrentar não apenas os desafios atuais da computação, mas também, e especialmente, aqueles que ainda estão por vir.

O investimento no aprendizado dos fundamentos da computação significa a garantia de supri-

mento de futuros bons profissionais para a área, assim como do desenvolvimento de boas e novas tecnologias e produtos.

A falta de investimentos, por outro lado, pode Vol. 1 acarretar prejuízos que vão bem além da falta de bons profissionais e das limitações decorrentes da sua atuação no mercado. Ela pode, no prazo de alguns anos, provocar uma escassez crítica de professores e pesquisadores com esse perfil, sem os quais toda e qualquer ação de recuperação na área ficaria definitivamente comprometida. Num mundo globalizado, ela pode também criar dependências de caráter científico que poderiam ser nocivas ao país.

A discussão acerca dos aspectos teóricos da computação, e da forma de ensino dos mesmos, deve ser, portanto, atual e permanente, não podendo ser relegada a um segundo plano, especialmente quando se percebe que as áreas de natureza tecnológica estão sempre, e cada vez mais, pressionando por mais disciplinas e mais carga horária nos cursos de graduação em computação.

3 Temas

O corpo de conhecimento contido na área de linguagens formais e autômatos é derivado de duas sub-áreas de conhecimento que eram consideradas de forma independente uma da outra até a década de 1960: linguagens formais, de um lado, e teoria de autômatos, de outro. Foi a partir das pesquisas efetuadas e dos resultados obtidos nessa década que se pôde estabelecer uma relação mais direta entre tais sub-áreas, as quais foram, a partir de então, combinadas numa única grande área de conhecimento e são, nos dias de hoje, consideradas como sendo praticamente indissociáveis uma da outra.

A sub-área linguagens formais trata da caracterização, classificação, formalização e estudo das propriedades das linguagens estruturadas em frases. O marco inicial dessa sub-área é o trabalho de Chomsky, um matemático e linguísta que, em meados da década de 1950, pesquisava a formalização gramatical de linguagens naturais. Apesar do seu insucesso no cumprimento desse objetivo, ele apresentou uma classificação das linguagens estruturadas em frases, organizada em níveis de complexidade crescente, que se tornaria referência fundamental para o estudo das linguagens formais. Tal classificação passou a ser conhecida como Hierarquia de Chomsky [2].

Partindo-se da classe de linguagens mais “simples” em direção à classe de linguagens mais “complexa” (os termos “simples” e “complexa” referem-se aos diversos aspectos das classes de linguagens consideradas, mas costumam englobar propriedades estruturais, aspectos de formalização sintática, custo do reconhecimento e grau de dificuldade na demonstração de certas propriedades),

a Hierarquia de Chomsky contempla: a classe das linguagens regulares (ou tipo 3); a classe das linguagens livres de contexto (ou tipo 2); a classe das linguagens sensíveis ao contexto (ou tipo 1) e, finalmente, a classe das linguagens recursivamente enumeráveis (ou tipo 0).

Por se tratar de uma taxonomia de linguagens na qual se percebe, naturalmente, uma escala de complexidade crescente, e, também, devido ao fato de que as classes de complexidade inferior estão propriamente contidas nas classes de complexidade mais elevada, é comum considerar a Hierarquia de Chomsky como ponto de partida para a organização de conteúdos programáticos (e outros materiais didáticos) para a disciplina linguagens formais e autômatos, adotando-se as diferentes classes de linguagens contidas nessa hierarquia como principais temas de estudo.

Tal organização proporciona ao aluno uma introdução natural aos assuntos da área e permite que o seu envolvimento aconteça de maneira gradual, à medida que o estudo das classes de linguagens mais simples dá lugar ao estudo das classes de linguagens mais complexas.

Como as classes de linguagens estão propriamente contidas umas nas outras, o estudo de uma nova classe a partir da anterior preserva todo o conhecimento adquirido pelo aluno até aquele momento, e permite o aprendizado incremental da matéria, em que são estudados apenas os aspectos específicos das linguagens contidas na nova classe e que não existiam na classe anterior.

Após a consolidação da Hierarquia de Chomsky como referencial principal para o estudo das linguagens formais, novas pesquisas identificaram diversas novas e importantes classes de linguagens situadas entre aquelas já identificadas por Chomsky. Entre as mais importantes encontram-se: a classe das linguagens livres de contexto determinísticas e a classe das linguagens recursivas.

Por se tratar de classes de linguagens relevantes tanto do ponto de vista teórico quanto prático, pelo menos um subconjunto dessas seis classes de linguagens (regulares, livres de contexto determinísticas, livres de contexto gerais, sensíveis ao contexto, recursivas e recursivamente enumeráveis) é normalmente adotado como base para a seleção dos temas principais que serão desenvolvidos numa disciplina sobre o assunto.

Se, por um lado, a sub-área de linguagens formais oferece uma perspectiva de estudo baseada na síntese de cadeias (através de dispositivos gramaticais) para o estudo das várias classes de linguagens, por outro lado a sub-área de teoria de autômatos oferece uma perspectiva de estudo baseada na análise de cadeias (através de dispositivos de reconhecimento) com o mesmo objetivo. Naturalmente, a equipotência dos dispositivos de síntese e de análise para cada classe de linguagens considerada permite uma grande liberdade

de escolha na forma de organização do material didático.

Os três tipos de tema que podem ser considerados como principais estruturadores de materiais didáticos sobre o assunto são, portanto: (i) as várias classes de linguagens; (ii) os vários tipos de dispositivos de análise e (iii) os vários tipos de dispositivos de síntese. Como consequência, é comum encontrar materiais didáticos que sejam estruturados em torno de combinações diversas desses três tipos de tema, sem preocupação especial com relação à sua uniformidade.

No caso de temas escolhidos no universo dos dispositivos de análise, as denominações tipicamente adotadas são “Autômatos Finitos” (dispositivos capazes de reconhecer linguagens regulares), “Autômatos de Pilha” (capazes de reconhecer linguagens livres de contexto) e “Máquinas de Turing” (capazes de reconhecer, com pequenas variações, linguagens sensíveis ao contexto, linguagens recursivas e linguagens recursivamente enumeráveis).

De maneira similar, temas como “Gramáticas Livres de Contexto” e “Gramáticas Sensíveis ao Contexto” podem ser encontrados em materiais organizados em função dos tipos de dispositivo de síntese. “Linguagens Regulares”, “Linguagens Livres de Contexto” etc são típicos para temas derivados das principais classes de linguagens conhecidas.

Organizar um conteúdo programático em torno de temas que correspondam aos nomes das classes de linguagens ou dos tipos dos dispositivos de síntese e análise correspondentes não constitui, por si só, um obstáculo ao aprendizado. Os problemas podem surgir quando se adota um enfoque misto, em que partes do material são designados a partir dos nomes das classes de linguagens e outros são designados a partir dos nomes dos correspondentes dispositivos de análise ou síntese.

Não é incomum encontrar casos, por exemplo, com a seguinte sequência de temas: (i) “Linguagens Regulares”, (ii) “Linguagens Livres de Contexto” e (iii) “Máquinas de Turing”. A decisão, nesse caso, é normalmente justificada pelo fato de que, através das Máquinas de Turing é possível definir e discorrer sobre três classes de linguagens, todas elas de complexidade superior à classe das linguagens livres de contexto.

Do ponto de vista pedagógico, esse enfoque cria dificuldades para o aluno, uma vez que os principais elementos de estruturação do texto (ou seja, os seus temas) possuem naturezas diferentes (classes de linguagens e dispositivos de reconhecimento) contribuindo assim para uma menor uniformidade na apresentação do conteúdo.

O recomendado, num caso como esse, seria manter o conceito “classe de linguagens” como sendo a natureza de todos os temas principais, procurando abstrair as várias versões da Máquina

de Turing conforme a classe de linguagens que estiver sendo discutida.

Assim, por exemplo, o tema sobre linguagens sensíveis ao contexto poderia ser desenvolvido sobre Máquinas de Turing com fita limitada, o tema sobre linguagens recursivas sobre Máquina de Turing que sempre param e, finalmente, o tema sobre linguagens recursivamente enumeráveis sobre Máquinas de Turing que podem ou não parar quando a cadeia de entrada não pertence à linguagem em questão.

Trata-se, sem dúvida, de um desenvolvimento um pouco mais trabalhoso, se comparado com o tradicional, no qual adota-se como ponto de partida a Máquina de Turing geral para depois criarem-se versões restritas, derivando-se delas as subclasses de linguagens correspondentes. A estratégia proposta, no entanto, garante uma absorção gradual e sempre justificada do conteúdo apresentado, sem mudanças bruscas dos conceitos fundamentais orientadores do estudo e sem antecipações de assuntos que são, no fundo, dispensáveis para discutir a classe de linguagens sob foco.

Situações similares a essa acontecem quando são escolhidos, como temas principais, termos de naturezas distintas que designam, além dos tipos já citados, outros tópicos relacionados às classes de linguagens, como por exemplo: (i) as suas propriedades gerais (questões decidíveis, propriedades de fechamento etc); (ii) as suas propriedades específicas (minimização de autômatos finitos, “pumping lemmas” etc); ou (iii) particularidades de uma determinada classe de linguagens (como é o caso da decidibilidade).

A superação desse obstáculo conceitual pode ser alcançada se for possível atingir um elevado grau de uniformização na denominação dos temas principais, considerando-se apenas conceitos de mesma natureza, de preferência conceitos de maior abrangência e que englobem outros que lhe são naturalmente subordinados ou especializados.

Sugere-se, como alternativa, que os temas principais sejam denominados a partir das classes de linguagens que se deseja apresentar, iniciando-se com a mais simples e terminando com a mais complexa. Todos os demais assuntos relacionados (dispositivos de síntese, dispositivos de análise, propriedades etc) podem facilmente ser acomodados dentro de uma estrutura de tópicos (doravante denominados subtemas) que contemple não apenas os aspectos genéricos mas também os aspectos específicos das classes de linguagens consideradas.

Dessa forma, o ponto de partida para o estudo será sempre uma classe de linguagens, e todos os detalhes a ela relacionados ficam dentro dela encapsulados e a ela subordinados. O sequenciamento de temas, nessa proposta, evita que o foco do leitor seja deslocado do conceito mais fundamental — a classe de linguagens que está sendo estudada — para outros conceitos de abrangên-

cia mais localizada e de natureza distinta, contribuindo assim para uma melhor compreensão do contexto no qual se desenvolve o conteúdo.

4 Subtemas

A escolha, a denominação e o sequenciamento dos subtemas que são discutidos dentro de cada tema são igualmente importantes para motivar o aluno, facilitar a sua contextualização em relação restante do assunto e orientar o seu estudo.

Quando se trata da organização interna dos temas de um material didático sobre linguagens formais e autômatos, não existe uma estrutura conceitual padrão que possa servir como referência para o seu desenvolvimento. Por outro lado, os subtemas que são discutidos dentro de cada tema são considerados clássicos e há pouca variação em relação ao que se encontra num ou noutro texto da área. Por esse motivo, é usual se trabalhar com grande liberdade na geração do conteúdo, porém nem sempre o resultado favorece a construção de uma visão, e portanto de um entendimento, estruturados por parte do aluno.

Considera-se, normalmente, que cada classe de linguagens tem as suas características e o seu próprio conjunto de subtemas que devem ser discutidos no contexto da própria classe. Pouca atenção costuma ser dada às possibilidades de construção de um roteiro que possa ser replicado para diferentes classes de linguagens, o que poderia contribuir para uma melhor percepção dos subtemas que são comuns e daqueles que são específicos para as diversas classes de linguagens consideradas.

Uma análise preliminar de tais subtemas, no entanto, revela que, para as classes de linguagens consideradas anteriormente, todos os seguintes subtemas são presentes e relevantes, podendo assim constituir o alicerce para uma organização mais estruturada do material didático, independentemente da classe de linguagens considerada (com uma ou outra exceção): (i) propriedades estruturais; (ii) formalização gramatical; (iii) modelo de reconhecimento; (iv) equivalência entre os formalismos de representação de linguagens; (v) principais propriedades; (vi) a relação da classe corrente com a classe de linguagens precedente (consideradas numa escala de complexidade crescente) e, finalmente, (vii) a demonstração da existência de linguagens que não pertencem à classe de linguagens corrente (como fator motivador para a apresentação da classe de linguagens seguinte).

Considere-se, por exemplo, a classe das linguagens livres de contexto. De acordo com o enfoque proposto, uma possível sequência de apresentação dos subtemas relacionados a essa classe de linguagens, e que ainda guardam relação com subtemas equivalentes para as outras classes de linguagens, seria: (i) balanceamento de termos (aninhamento); (ii) gramáticas livres de contexto; (iii)

autômatos de pilha; (iv) equipotência dos autômatos de pilha com as gramáticas livres de contexto (no que se refere à capacidade de representação de linguagens livres de contexto); (v) questões decidíveis e propriedades de fechamento (das linguagens livres de contexto); (vi) demonstração de que a classe das linguagens regulares está propriamente contida na classe das linguagens livres de contexto e (vii) demonstração de que existem linguagens que não são livres de contexto (através do “Pumping Lemma” para as linguagens livres de contexto).

De maneira semelhante, roteiros equivalentes podem ser desenvolvidos para as demais classes de linguagens.

Se, por um lado, a escolha de um conjunto padrão de subtemas a ser desenvolvido para cada classe de linguagens colabora de maneira significativa para o melhor entendimento acerca da natureza de cada subtema e também da maneira como ele é contextualizado em cada classe de linguagens, por outro lado é fato que existem outros subtemas que são específicos de um grupo reduzido de classes de linguagens, ou mesmo pertinentes a uma única classe de linguagens.

Pode-se citar como exemplo, entre outros, os subtemas transdutores finitos e a minimização de autômatos finitos (para a classe das linguagens regulares), árvores de derivação e ambiguidade (para a classe das linguagens livres de contexto) e decidibilidade (para a classe das linguagens recursivamente enumeráveis).

Nesses casos, a harmonização de tais subtemas específicos, juntamente com os subtemas que sejam comuns a todas as demais classes de linguagens, pode produzir o efeito de fazer com que o aluno tenha uma melhor percepção das particularidades encontradas em cada classe de linguagens estudada, ou seja, das características e propriedades que são únicas em cada classe (ou subconjunto de classes).

Tal organização favorece, do ponto de vista pedagógico, uma compreensão mais clara acerca da natureza dos subtemas que são discutidos para cada classe de linguagens, e instiga o aluno a desenvolver mecanismos mentais de entendimento que sejam similares a outros previamente construídos, ou totalmente novos e independentes, conforme os subtemas estudados possuam naturezas idênticas ou totalmente distintas de outros anteriormente estudados.

Assim, por exemplo, acontece com o estudo dos “Pumping Lemma”. O mecanismo mental desenvolvido pelo aluno para compreender o “Pumping Lemma” para as linguagens regulares é semelhante ao que ele desenvolve para compreender o “Pumping Lemma” para as linguagens livres de contexto (finitude do número de estados em um autômato finito versus finitude do número de símbolos não-terminais em uma gramática livre de contexto).

Situação semelhante acontece com o estudo das formas normais para gramáticas, não importando se se trata da classe das linguagens regulares ou ainda das classes das linguagens livres de contexto, sensíveis ao contexto ou recursivamente enumeráveis.

Essa proposta evidencia, por outro lado, a inexistência de um “Pumping Lemma” para a classe das linguagens sensíveis ao contexto e, também, a inexistência de um subtema que discuta a minimização de autômatos de pilha. Isso, do ponto de vista pedagógico, pode ser bem explorado pelo professor e também pelo aluno.

Sugere-se, portanto, que a escolha e a ordenação dos subtemas a serem discutidos no contexto de cada classe de linguagens seja feita de forma que, num primeiro momento, se estabeleça uma correlação direta com subtemas de mesma natureza quando estes ocorrerem em classes diferentes de linguagens e, num segundo momento, se evidencie os subtemas que são específicos para uma ou mais classes de linguagens.

Através da percepção das semelhanças e também das diferenças entre os subtemas que compõem o corpo de conhecimento acerca de cada classe de linguagens, o aluno é instrumentado de forma que ele possa, em cada momento do seu estudo, compreender de forma contextualizada a relevância e a abrangência de cada tópico que lhe é apresentado, o que naturalmente contribui para uma melhor absorção do conteúdo.

5 Demonstrações

Não é raro encontrar materiais didáticos que deixam para o leitor a tarefa, nem sempre fácil, de completar a demonstração de certos teoremas.

Tal recurso pode ser considerado didático quando a disciplina prepara o aluno, até o ponto em que a exigência é apresentada, com as ferramentas necessárias à obtenção de uma solução que seja possível de ser alcançada de forma individual e sem esforços exagerados.

Além disso, há que se considerar a própria natureza dos teoremas cujas demonstrações são entregues ao aluno. Quando se trata de teoremas menores, secundários no desenvolvimento do conteúdo, ou mesmo de teoremas cuja demonstração é de baixa complexidade, não há dúvida de que esse é um recurso de aprendizagem válido.

Nesses casos, ele pode ser considerado como instrumento de motivação para o aluno e é importante como mecanismo pedagógico.

Quando, por outro lado, as demonstrações pedidas ao aluno são relativas a teoremas de grande importância no desenvolvimento da disciplina, ou, ainda, quando elas necessitam um instrumental e uma bagagem normalmente não encontradas em estudantes que se deparam pela pri-

meira vez com a matéria, então o efeito obtido será justamente o contrário. O aluno incapaz de efetuar as demonstrações solicitadas sente-se frustrado e, naturalmente, desmotivado a seguir em frente com os seus estudos.

A fim de se evitar a ocorrência de problemas de aprendizagem como consequência de demonstrações incompletas ou mesmo inexistentes, convém que todos os principais teoremas sejam demonstrados no próprio material didático ou no decorrer das atividades de ensino.

Como exemplos de teoremas que costumam ter a sua demonstração relegadas (pelo menos parcialmente) a um segundo plano, pode-se citar a equipotência plena (portanto bidirecional) entre cada um dos principais formalismos utilizados na representação de linguagens regulares (gramáticas lineares, autômatos finitos e expressões regulares), a demonstração da existência da forma normal $\alpha A \beta \rightarrow \alpha \gamma \beta$ para gramáticas sensíveis ao contexto, a demonstração da equivalência entre a classe das linguagens aceitas por Máquinas de Turing com fita ilimitada e daquelas geradas pelas gramáticas irrestritas e ainda a demonstração da existência de linguagens estruturadas em frases que não são recursivamente enumeráveis.

6 Intuição e redundâncias

A aquisição de um conhecimento é uma tarefa complexa que pode ser simplificada quando se faz uso de dois mecanismos complementares importantes: explicações que apelam para a intuição do aluno e o emprego de formas alternativas para se apresentar um mesmo assunto ou resultado.

Os materiais didáticos da área de linguagens formais e autômatos adotam, normalmente, uma linguagem matemática (e portanto árida), que requer grande capacidade de abstração e raciocínio lógico do aluno. Se, por um lado, essa característica não pode ser mudada na sua essência (pois a matemática, especialmente a matemática discreta, é a base para todo o corpo de conhecimento desenvolvido acerca do assunto), por outro lado é fato que, por trás de todo formalismo, dispositivo algébrico ou demonstração de algum teorema, existe uma fundamentação que pode ser traduzida em termos mais coloquiais, sem prejuízo para o rigor matemático necessário ao seu desenvolvimento, e que colabora de forma significativa para uma melhor compreensão dos assuntos que estão sendo apresentados.

Dessa maneira, constitui boa prática acrescentar, ao lado das demonstrações matemáticas formais, textos em linguagem natural, exemplos que de alguma forma remetam às experiências ou conhecimentos prévios do aluno, a comparações com resultados ou assuntos similares encontrados na mesma área ou em outras áreas do conhecimento, e o que mais se julgar conveniente, de modo que,

juntamente com o rigor dos formalismos empregados, o aluno tenha mais e melhores condições de inferir sobre o que se pretende demonstrar, fixando com mais convicção os resultados e despertando para correlações nem sempre óbvias que possam existir entre o tópico estudado e outros que façam parte do mesmo corpo de conhecimento ou mesmo de disciplinas distintas.

O emprego de redundâncias também contribui com esses objetivos. Certos teoremas podem ser demonstrados através da análise de um ou mais algoritmos que produzem o resultado desejado. Por exemplo, a eliminação de transições em vazio em autômatos finitos, a minimização de autômatos finitos e a demonstração da equipotência entre autômatos de pilha não-determinísticos e gramáticas livres de contexto podem ser feitas através de mais de um algoritmo diferente em cada caso.

Nessas situações, a escolha de apenas um algoritmo em cada caso suprime, para o aluno, a possibilidade de comparar soluções, compreender melhor a natureza de cada uma delas e, finalmente, considerar a possibilidade de desenvolver a sua própria solução.

O uso de redundâncias é um recurso didático que pode ser decisivo e ajudar muito o aluno que, por um motivo qualquer, às vezes enfrenta obstáculos de difícil transposição se o caminho que lhe é oferecido é único. Nem sempre o que é considerado mais claro, mais simples ou mais didático pelo professor é percebido da mesma forma pelo aluno, uma vez que tais adjetivos não possuem um sentido objetivo e, geralmente, o seu significado decorre das experiências prévias vividas pelo indivíduo em questão.

Não obstante, não é raro encontrar desenvolvimentos, ou mesmo materiais didáticos, nos quais a complexidade algébrica e dos formalismos é relegada a um segundo plano, e a apresentação da teoria é reduzida a uma mera enumeração de resultados, ou pouco mais do que isso. Tal estratégia prejudica a formação de bons profissionais, e também de futuros pesquisadores, e não deve ser incentivada.

7 Temas relacionados

Certos tópicos não são direta ou obviamente relacionados. Em alguns casos, no entanto, um enfoque que desvende relações menos óbvias entre esses tópicos, relacionando-os de alguma forma não usual ou não tradicional, pode contribuir, e muito, para um melhor entendimento de ambos os assuntos, eventualmente alargando as fronteiras de visão do aluno para além do trivial.

Isso acontece, por exemplo, com as questões decidíveis das linguagens livres de contexto e quando a questão “A cadeia w pertence à linguagem L ”? é apresentada ao aluno. O encaminhamento natural da resposta para essa questão consiste na apresen-

tação de um algoritmo que, aceitando como entrada uma gramática livre de contexto na Formal Normal de Greibach, gera sempre como saída uma resposta (afirmativa ou negativa) para a questão acima.

Se essa estratégia contempla uma solução satisfatória para a questão apresentada, por outro lado é fato que, nesse estágio do estudo, a inserção de tópicos contendo uma introdução ao tema da análise sintática é bem-vindas, pois eles guardam uma relação direta com a questão formulada.

Se adotada, essa estratégia permite que o aluno compreenda melhor a abrangência e a profundidade de uma questão aparentemente simples e sem maiores consequências, e, além disso, entenda a origem de toda uma tecnologia que foi desenvolvida com o objetivo de produzir soluções eficientes para a resposta dessa mesma pergunta. Em adição, podese introduzir as classes das linguagens $LL(k)$ e $LR(k)$, mostrando a relação delas com as demais classes de linguagens consideradas, em particular a classe das linguagens livres de contexto. Como resultado, verifica-se que a exploração conjunta dessas dois tópicos contribui para fortalecer a compreensão de ambos.

Outra situação que pode ser explorada de forma semelhante refere-se ao estudo dos temas ambiguidade e não-determinismo em linguagens livres de contexto, dois assuntos encontrados normalmente na maioria dos materiais didáticos da área de linguagens formais. A exploração da relação entre eles, no entanto, não é encontrada com tanta facilidade.

A definição de gramáticas e de linguagens ambíguas, em particular de linguagens inerentemente ambíguas (ou seja, aquelas para as quais não existem gramáticas que não sejam ambíguas), conduz naturalmente a um particionamento da classe das linguagens livres de contexto entre (i) o subconjunto das linguagens que são inerentemente ambíguas e (ii) o subconjunto das linguagens que são, por assim dizer, não-ambíguas (ou seja, aquelas para as quais é possível identificar pelo menos uma gramática que as gerem e que não sejam ambíguas).

Por outro lado, a definição de determinismo também conduz a uma partição da classe das linguagens livres de contexto. Nesse caso, entre o subconjunto das linguagens para as quais é possível identificar pelo menos um autômato de pilha determinístico que as reconheça (as chamadas linguagens livres de contexto determinísticas) e o subconjunto das linguagens para as quais isso não é possível (ou seja, linguagens que podem ser reconhecidas apenas através de autômatos de pilha não-determinísticos), também conhecida como classe das linguagens livres de contexto não-determinísticas.

Determinismo (ou não-determinismo) e ambiguidade (ou não-ambiguidade) são dois conceitos

que ensejam duas partições da classe das linguagens livres de contexto. E a pergunta que surge naturalmente é: “Qual é a relação que existe entre essas partições?”.

Uma investigação um pouco mais cuidadosa [3] permite concluir, no entanto, que a classe das linguagens livres de contexto determinísticas está propriamente contida na classe das linguagens livres de contexto não-ambíguas. Ou seja, (i) toda linguagem determinística é também não-ambígua; (ii) nem toda linguagem não-ambígua pode ser reconhecida de forma determinística e (iii) existem linguagens que são, simultaneamente, inerentemente ambíguas e não-determinísticas.

Portanto, a classe das linguagens livres de contexto pode ser considerada a partir de três outras subclasses propriamente contidas umas nas outras, e todas elas de grande importância prática e teórica: a subclasse das linguagens determinísticas descendentes $LL(k)$, a subclasse das linguagens determinísticas ascendentes $LR(k)$, que corresponde também à classe das linguagens livres de contexto determinísticas, e a subclasse das linguagens não-ambíguas, esta última contida propriamente na classe das linguagens livres de contexto gerais.

Tal enfoque, não usualmente adotada no ensino de linguagens formais e autômatos, associa dois conceitos aparentemente desconexos (determinismo e ambiguidade) e produz resultados que fazem com que o leitor tenha melhor domínio e compreensão sobre ambos, entendendo vários dos seus desdobramentos, como por exemplo a definição natural de outras classes de linguagens, além das inicialmente consideradas. A alternativa (ou seja, a não exploração dessa correlação) simplesmente deixa de contribuir para uma maior solidez e abrangência do aprendizado do aluno.

8 Pesquisas anteriores e atuais

A área de linguagens formais e autômatos desenvolveu-se principalmente nas décadas de 1960 e 1970, quando pesquisadores no mundo inteiro voltaram as suas atenções para as questões em aberto da área, conseguindo produzir os resultados mais importantes que são conhecidos e usados nos dias hoje. Por esse motivo, a área é entendida como sendo formada por um corpo de conhecimento sólido e estável, em torno do qual pouco se pode fazer em termos de inovação, inclusive e principalmente do ponto de vista pedagógico.

Isso, no entanto, não precisa ser necessariamente verdade. Tal visão desconsidera os seguintes fatos: (i) que parte do grande volume de conhecimento produzido nos anos passados não está presente de forma significativa nos conteúdos programáticos e nos materiais didáticos atuais; e (ii) que os resultados das pesquisas atuais não são se-

quer citados nesses mesmos materiais.

Se, por um lado, é fato que os conhecimentos que são gerados ao longo de anos de pesquisa devem sempre ser relativizados, priorizados, depurados e sintetizados, para dessa forma compor uma disciplina moderna e consistente, formada apenas por material relevante para a teoria e/ou para a prática profissional, por outro lado não se pode deixar de observar que muitos dos tópicos que hoje são considerados ultrapassados na área ainda podem ter um papel importante como recurso didático, com grande benefícios para o aprendizado de linguagens formais e autômatos.

Um exemplo disso são os estudos que foram feitos acerca de gramáticas com derivações controladas [4] (como é o caso das gramáticas controladas por linguagens regulares, das gramáticas matriciais, das gramáticas programadas ou ainda das gramáticas periodicamente variantes no tempo). Trata-se de formalismos gramaticais livres de contexto que, através de algum tipo de mecanismo de extensão relativamente simples, tornam-se capazes de representar linguagens sensíveis ao contexto.

De especial relevância é o caso das gramáticas livres de contexto com derivações controladas por linguagens regulares. Do ponto de vista didático, trata-se de dois formalismos clássicos, que são estudados em qualquer curso tradicional de linguagens formais e autômatos. No entanto, uma simples combinação desses formalismos resulta num terceiro e novo formalismo, potente o suficiente para representar linguagens de uma classe hierarquicamente superior, sem, portanto, a necessidade de se recorrer a formalismos mais complexos como é o caso das gramáticas de atributos, das gramáticas de dois níveis ou mesmo das gramáticas sensíveis ao contexto tradicionais que, em qualquer caso, apresentam um nível de sofisticação que desestimula o aluno.

Ao resgatar e introduzir assuntos antigos nos conteúdos programáticos atuais, como é o caso das gramáticas com derivações controladas, o aluno é favorecido em vários aspectos. Ele: (i) percebe como simples combinação de outros formalismos já conhecidos pode resultar em novos formalismos mais poderosos, e se sente estimulado a raciocinar de forma similar em outros contextos; (ii) constrói a sua compreensão de um assunto novo (no caso, a representação sintática formal de linguagens sensíveis ao contexto) a partir da compreensão de formalismos previamente estudados e de complexidade relativamente simples. Apesar da pouca divulgação nas salas de aula, e também nos materiais didáticos de apoio produzidos, existem muitos resultados de pesquisa atual na área que, se forem apresentados de forma comedida, apoiados em fundamentações de clara percepção pelo aluno e relacionados com os tópicos clássicos da disciplina, podem ser explorados sem grandes dificuldades no ambiente de aprendizagem, com gran-

des vantagens para o aluno.

É representativo o caso dos autômatos de pilha estruturados [5], formalismo desenvolvido para permitir a construção de reconhecedores sintáticos que apresentam vantagens importantes, inclusive didáticas, quando comparados com os tradicionais autômatos de pilha.

Organizado na forma de múltiplas submáquinas mutuamente recursivas, esse formalismo apresenta, como principais vantagens: (i) uma maior eficiência de execução (pois a pilha é utilizada apenas quando isso for estritamente necessário, ou seja, no reconhecimento de construções aninhadas, operando como um autômato finito quando esse não for o caso), e (ii) a possibilidade de se criar projetos e implementações modulares, com ganhos em termos de legibilidade e de manutenção.

Um outro exemplo são as pesquisas atuais sobre dispositivos adaptativos [6], as quais podem ser aplicadas a dispositivos de natureza diversa, como é o caso das gramáticas e dos autômatos.

A inserção de tópicos sobre gramáticas e autômatos adaptativos num texto introdutório sobre linguagens formais e autômatos pode trazer os seguintes benefícios para o aluno: (i) demonstra, como nos casos das gramáticas controladas por regras e dos autômatos de pilha estruturados, que a concepção de novos formalismos com base nos formalismos tradicionais pode ser vantajosa sob inúmeros aspectos, especialmente os de caráter mais prático (facilidade de construção e eficiência da operação); (ii) demonstra que a área ainda é objeto de pesquisas e que ela constitui, portanto, campo fértil para a atuação de futuros jovens pesquisadores.

Naturalmente que a idéia não é esgotar todos os principais resultados de uma linha de pesquisa atual (qualquer que seja ela) num material ou numa atividade didática de caráter introdutório, até porque esse não é o seu objetivo principal. No entanto, a simples citação, apresentação e exemplificação de alguns resultados mais importantes de uma linha de pesquisa moderna pode ter um efeito altamente motivador sobre o aluno, criando perspectivas positivas para o aprofundamento do seu estudo e propiciando uma melhor compreensão dos conceitos da teoria clássica.

Recomenda-se, portanto, manter sempre em vista o conhecimento gerado no passado, muitas vezes deixado perdido no tempo e no espaço, mas ainda assim capaz de contribuir para uma aprendizagem mais eficaz no momento atual, e também o conhecimento gerado no presente, muitas vezes de circulação restrita, mas, apesar disso, maduro o suficiente para se relacionar com o conhecimento tradicional, proporcionando ao aluno mais e melhores instrumentos de aprendizagem.

9 Experimentação

É comum que os materiais e as atividades didáticas de linguagens formais e autômatos adotem um enfoque essencialmente algébrico, o que exige dos alunos não apenas uma boa formação matemática mas também, e principalmente, uma grande capacidade de raciocínio lógico e abstrato. Por esse motivo, os estudantes de tais textos precisam lidar com um nível de complexidade adicional, além daquele que é inerente aos assuntos da área, para conseguir bons resultados na disciplina.

Não é raro, por outro lado, encontrar enfoques que buscam superar essas dificuldades através do uso de ferramentas computacionais que permitem ao aluno vivenciar, em ambientes de simulação, os principais conceitos vistos na teoria. Ao contemplar o estudante com uma estratégia didática complementar e alternativa em relação àquela exclusivamente algébrica, criam-se as condições para que, através de perspectivas diferentes, ele obtenha um nível de compreensão mais completo em relação à matéria.

A categoria mais comum de dispositivos empregados para essa finalidade são os simuladores, sejam eles gráficos ou não. Um simulador é um programa de computador que permite ao aluno construir dispositivos computacionais que reproduzam comportamentos equivalentes aos dos dispositivos algébricos estudados na teoria.

Dessa forma, é possível experimentar com gramáticas, reconhecedores, transdutores ou outros formalismos abstratos (como por exemplo expressões regulares), bem como verificar suas propriedades (por exemplo, determinar se a linguagem aceita por um autômato finito é vazia ou não) ou aplicar algoritmos de mapeamento (por exemplo para a minimização de autômatos finitos).

A diferença entre o uso de um ambiente gráfico ou não-gráfico (baseado em texto apenas) consiste na facilidade com que o usuário consegue criar, visualizar e modificar os seus modelos para, em seguida, aplicar sobre eles um conjunto normalmente básico e predefinido de operações disponibilizadas pelo simulador.

Um simulador de gramáticas livres de contexto poderia, por exemplo, além de verificar a sintaxe da gramática fornecida como entrada, oferecer ao usuário operações que lhe permitissem, por exemplo, gerar de forma aleatória sentenças pertencentes à linguagem por ela definida, montar árvores de derivação, detectar a existência de símbolos inúteis e inacessíveis, simplificar a gramática e obter uma forma normal qualquer equivalente. Conforme o tipo de formalismo, a lista pode ser tornar relativamente extensa.

De especial interesse, quando se trata da simulação de autômatos (finitos, de pilha, Máquinas de Turing, etc), são os simuladores gráficos, que permitem ao usuário um grande conforto na edi-

ção e visualização dos dispositivos. Através de menus, janelas e botões, e usando principalmente as ações de clicar e arrastar com o mouse, o usuário consegue facilmente construir o seu dispositivo, deixando-o pronto para a aplicação das operações disponibilizadas pela ferramenta. São diversas as ferramentas existentes com esse objetivo, mas talvez a mais conhecida seja o JFLAP [7].

Os simuladores (sejam eles gráficos ou não) servem como importante instrumento de aprendizagem e contribuem de forma decisiva para uma melhor compreensão dos aspectos dinâmicos dos dispositivos estudados (como, por exemplo, a transição entre estados com consumo de símbolo num autômato finito, ou a derivação de uma forma sentencial numa gramática), e servem também para a visualização do resultado da aplicação prática de algoritmos de transformação ou de mapeamento de dispositivos (como por exemplo a eliminação de não-determinismos em autômatos finitos ou a obtenção de expressões regulares a partir de autômatos finitos).

Seja como auxílio para analisar um comportamento de um dispositivo ou de um método, seja como recurso para validar as respostas de exercícios propostos ou ainda como ferramenta para exploração de conceitos obscuros da teoria, o fato é que tais simuladores oferecem ajuda significativa para o aluno que, de outra forma, estaria limitado às representações algébricas e à sua própria capacidade de raciocínio e de abstração.

Uma outra vantagem derivada dessa estratégia consiste no fato de que o aluno de computação passa a utilizar uma ferramenta computacional para estudar a matéria, o que normalmente serve como elemento de estímulo para o seu aprendizado.

Apesar de todas essas vantagens, tais simuladores limitam a capacidade de experimentação do usuário final, a qual costuma acontecer apenas dentro das possibilidades previstas pelo desenvolvedor da própria ferramenta.

Assim, por exemplo, o usuário encontra à sua disposição um conjunto limitado e fechado de dispositivos sobre os quais pode realizar simulações. Além disso, para cada tipo de dispositivo previsto, ele pode realizar apenas os experimentos preconcebidos, devendo se satisfazer com os mesmos. Suponha-se, como exemplo, que um certo simulador permita ao usuário construir modelos de linguagens regulares.

Suponha-se também que ele ofereça ainda uma série de operações sobre tais modelos. Mas imagine-se que o usuário deseja experimentar com questões decidíveis das linguagens regulares, em particular sobre linguagens que sejam vazias, finitas ou infinitas. Se a ferramenta não oferecer suporte para esse tipo de experimentação, nada poderá ser feito no âmbito da simulação e o usuário deverá se contentar com a forma tradicional

para a fixação do conteúdo.

Tal situação decorre do fato de que a maioria dos simuladores existentes atualmente são fechados, ou seja, o usuário não dispõe de condições para modificar ou estender o seu comportamento, agregando novas funcionalidades ou alterando outras características.

Como alternativa, sugere-se o uso de ambientes de simulação abertos (em contraposição aos “simuladores fechados”), nos quais o usuário, além de poder utilizar os seus modelos predefinidos, experimentando-os à luz de um conjunto de operações também predefinidas, pode ainda: (i) criar novas operações para modelos predefinidos; (ii) modificar operações preexistentes para modelos predefinidos e (iii) definir novos modelos de dispositivo, atribuindo-lhes um conjunto de operações iniciais.

Simuladores com essas características podem ainda beneficiar os seus usuários na medida em que os modelos e operações que constituam o seu núcleo inicial façam parte de um projeto estruturado, coerente e bem documentado, que permita especializações e reaproveitamento de código com facilidade e naturalidade. Dessa forma, o usuário compreende melhor o projeto do conjunto de modelos iniciais, entende a sua fundamentação e as suas relações internas e externas, e percebe as possibilidades de desenvolvimento que eles oferecem, sentindo-se estimulado para criar novos modelos e novas operações baseadas nas já existentes.

Tal estratégia serve, em última instância, para que alunos e professores possam aprender não apenas a partir de um conjunto predefinido de modelos e operações, mas também, e principalmente, a partir do projeto, implementação e testes de novos modelos e das suas respectivas operações. Cria-se, assim, a possibilidade estimulante de se trabalhar com projetos e atividades de laboratório numa disciplina de caráter essencialmente teórico.

Não bastassem essas vantagens, a estratégia proposta permite a exploração de um outro aspecto, de caráter mais prático, e portanto de especial interesse para os cursos de engenharia de computação ou afins. Trata-se da questão da implementação dos modelos teóricos e das diferentes técnicas que podem ser aplicadas nessa importante etapa, levando-se em conta os resultados esperados de performance e outros relacionados.

Por esse motivo, o uso de simuladores abertos pode ser visto, também, como uma estratégia que ajuda a promover o interrelacionamento de uma disciplina de caráter principalmente teórico (linguagens formais e autômatos) com outras disciplinas da computação, de caráter mais prático e, muitas vezes, já conhecidas e dominadas pelos alunos (como é o caso da modelagem, metodologia de desenvolvimento, programação e testes de software), o que (i) contribui para diminuir o isolamento da disciplina no contexto do curso; (ii) serve

como estímulo para o aluno, na medida em que ele pode relacionar o conteúdo da mesma com o de outras disciplinas; (iii) permite a experimentação associada à realização de projetos e à construção de artefatos de software.

Apesar de todas as vantagens apresentadas, o emprego de simuladores, sejam eles gráficos ou não, fechados ou abertos, deve sempre ser feito de modo criterioso, a fim de não preterir o pensamento algébrico do aluno. Se usadas de forma ampla e sem planejamento, tais ferramentas podem induzir uma espécie de preguiça mental que prejudica o alcance dos objetivos da disciplina.

10 Estudos de caso

A organização da disciplina Linguagens Formais e Autômatos em torno de temas e subtemas que apresentem uma certa uniformidade e, especialmente, que sejam organizados de forma hierárquica, permitindo ao aluno identificar afinidades entre eles, pode ser, conforme visto nas seções anteriores, um grande facilitador do aprendizado. Na prática, no entanto, as principais referências de estruturação curricular para essa disciplina não refletem tal visão.

Um exemplo disso pode ser encontrado no “Computing Curricula 2001 — Computer Science” [8], uma referência da ACM (Association for Computing Machinery) para a elaboração de currículos para cursos de Ciência da Computação.

Nesse documento, o tópico Teoria de Autômatos é definido dentro da área Algoritmos e Complexidade, e possui a seguinte relação de unidades sugerida: (i) autômatos finitos determinísticos; (ii) autômatos finitos não-determinísticos; (iii) equivalência entre autômatos finitos determinísticos e não-determinísticos; (iv) expressões regulares; (v) “pumping lemma” para as expressões regulares; (vi) autômatos de pilha; (vii) relação entre autômatos de pilha e gramáticas livres de contexto; (viii) propriedades de gramáticas livres de contexto; (ix) Máquinas de Turing; (x) Máquinas de Turing não-determinísticas; (xi) conjuntos e linguagens; (xii) Hierarquia de Chomsky; e (xiii) Tese de Church-Turing.

As unidades (i), (ii), (vi), (ix) e (x) referem-se a modelos de dispositivos de análise de linguagens. A unidade (iv) refere-se a um dispositivo de síntese de linguagens. As unidades (iii) e (vii) referem-se à equipotência entre dispositivos de naturezas distintas. As unidades (v) e (viii) discutem propriedades das correspondentes classes de linguagens. E, finalmente, as unidades (xi), (xii) e (xiii) discutem tópicos isolados. Não há, portanto, uma uniformidade conceitual significativa no conjunto de unidades sugeridas.

Outro exemplo é encontrado no “Currículo de Referência da SBC para Cursos de Graduação em Computação e Informática” [1]. Nele, a matéria

Linguagens Formais, Autômatos e Computabilidade está contida no núcleo denominado Fundamentos da Computação, e possui os seguintes tópicos sugeridos: (i) gramáticas; (ii) linguagens regulares, livres-de-contexto e sensíveis-ao-contexto; (iii) tipos de reconhecedores; (iv) operações com linguagens; (v) propriedades das linguagens; (vi) autômatos de estados finitos; (vii) autômatos de pilha; (viii) Máquina de Turing; (ix) funções recursivas; (x) Tese de Church; (xi) Teorema da Incompletude de Gödel; (xii) classes de problemas P, NP, NP-completo e NP-difícil; (xiii) métodos de redução de problemas.

Aqui também é possível observar o uso de temas de natureza distinta para definir o conteúdo da matéria. Senão, os tópicos (iii), (vi), (vii) e (viii) referem-se a dispositivos de análise, e os tópicos (i), (ii), (iv) e (v), respectivamente, a dispositivos de síntese, classes de linguagens, operações e propriedades das classes de linguagens.

É claro que uma simples relação de tópicos, como essas da ACM e da SBC, não precisa, necessariamente, ser adotada como roteiro definitivo para a elaboração de conteúdos programáticos ou de materiais didáticos. Mas é provável que uma organização de tópicos alternativa, compatível com os princípios anteriormente apresentados, talvez represente uma contribuição importante para aqueles que buscam em tais documentos referências para a preparação dos seus trabalhos.

Também nos livros-texto da área encontram-se situações semelhantes. Esse é o caso de [9], considerado, até os dias hoje, umas das principais referências sobre o assunto.

Seus capítulos são: (i) preliminares; (ii) autômatos finitos e expressões regulares; (iii) propriedades dos conjuntos regulares; (iv) gramáticas livres de contexto; (v) autômatos de pilha; (vi) propriedades das linguagens livres de contexto; (vii) Máquinas de Turing; (viii) indecidibilidade; (ix) Hierarquia de Chomsky; (x) linguagens livres de contexto determinísticas; (xi) propriedades de fechamento das famílias de linguagens. O livro segue com outros capítulos.

Além de comentários similares aos feitos anteriormente para os currículos de referência da ACM e da SBC, é possível fazer alguns comentários adicionais acerca dessa organização.

Os capítulos (ii) e (iii) tratam da classe das linguagens regulares, mas isso não fica explícito através de uma simples inspeção do sumário. O mesmo ocorre com os capítulos (iv), (v), (vi) e (x), que tratam da classe das linguagens livres de contexto. No caso do capítulo (ii), os dispositivos de síntese e análise são discutidos de forma conjunta. No caso das linguagens livres de contexto, eles são discutidos em capítulos separados (respectivamente os capítulos (iv) e (v)).

Já o capítulo (vii) serve para introduzir as Má-

quinas de Turing e as suas principais variações, ao mesmo tempo em que faz uso das mesmas para definir e apresentar as classes das linguagens recursivas e recursivamente enumeráveis. A discussão da importante classe das linguagens sensíveis ao contexto, por outro lado, que também pode ser feita a partir da definição de um modelo restrito de Máquina de Turing, é apresentada apenas no capítulo (xi), onde se discute a Hierarquia de Chomsky.

Situações semelhantes podem ser encontradas em livros-texto mais modernos, como é o caso de [10], que possui a seguinte estrutura de capítulos: (i) preliminares da matemática; (ii) linguagens; (iii) gramáticas livres de contexto; (iv) formas normais para gramáticas livres de contexto; (v) autômatos finitos; (vi) propriedades das linguagens regulares; (vii) autômatos de pilha e linguagens livres de contexto; e (viii) Máquinas de Turing. O livro segue com outros capítulos.

Após uma introdução genérica ao assunto linguagens (no capítulo (ii)), o autor discute gramáticas livres de contexto e formas normais para gramáticas livres de contexto nos capítulos (iii) e (vi), respectivamente. Na sequência, são discutidos os autômatos finitos (capítulo (v)) e as propriedades das linguagens regulares (capítulo (vi)). Finalmente, o capítulo (vii) retoma a discussão acerca das linguagens livres de contexto e dos autômatos de pilha.

Percebe-se, portanto, que o texto inicia com a apresentação indireta de uma classe de linguagens de nível intermediário (a classe das linguagens livres de contexto), onde é discutido o respectivo modelo de síntese e, em seguida, são apresentadas as correspondentes formas normais. Logo depois é a vez da classe das linguagens regulares, analisadas inicialmente pela perspectiva do correspondente dispositivo de análise e depois pela discussão das suas principais propriedades. Por último, o capítulo (vii) retoma a discussão da classe das linguagens livres de contexto, e os tópicos, dessa vez, são os seus dispositivos de análise e a própria classe, enquanto categoria de linguagens.

As discussões sobre as gramáticas regulares e as suas respectivas formas normais, assim como sobre as principais propriedades da classe das linguagens livres de contexto, ficam confinadas, como subtemas de difícil localização, dentro dos temas acima indicados.

Comentário idêntico ao que foi feito para a referência [9], no caso das Máquinas de Turing e das classes das linguagens recursivas, recursivamente enumeráveis e sensíveis ao contexto, também pode ser feito para a referência [10].

Por isso, apresenta-se a seguinte sugestão de tópicos como alternativa para a organização de aulas e de textos didáticos sobre linguagens formais e autômatos: (i) conceitos básicos de linguagens (símbolo, alfabeto, cadeias e linguagens); (ii) modelos genéricos de síntese (gramáticas) e análise

(reconhecedores) de linguagens; (iii) Hierarquia de Chomsky; e (vi) classes de linguagens (regulares, livres de contexto, sensíveis ao contexto, recursivas e recursivamente enumeráveis), seus modelos de síntese e análise, a relação entre as classes e suas principais propriedades. Para cada uma das classes de linguagens relacionadas em (iv), pode-se considerar a estrutura de subtemas sugerida na seção 3 do presente artigo.

Essa sugestão privilegia as classes de linguagens como principais elementos estruturadores do conteúdo, e desenvolve, para cada uma delas, os tópicos que lhe são naturalmente relacionados.

11 Conclusão e resumo

O presente artigo apresenta uma série de ideias e sugestões com a intenção de contribuir para a criação de conteúdos programáticos e de materiais didáticos originais sobre linguagens formais e autômatos e o ensino dessa mesma disciplina.

Essas ideias e sugestões passam inicialmente pela estruturação do corpo de conhecimento da área, enfatizando a necessidade de construí-lo acerca de temas de mesma natureza. Considera-se, em seguida, o detalhamento de cada grande tema da área, identificando, inicialmente, os subtemas que sejam comuns a todos os temas e, posteriormente, os subtemas que sejam particulares a um ou mais temas.

O desenvolvimento de cada subtema pode, por sua vez, se beneficiar, do ponto de vista pedagógico, da exploração de tópicos de vaga correlação aparente, do uso de argumentos intuitivos, do emprego de redundâncias e da demonstração de todos os resultados considerados fundamentais para se apresentar os principais resultados, sem relegar tal tarefa para o aluno.

O uso do resultado de pesquisas anteriores, eventualmente deixado em desuso nos dias de hoje por questões de natureza não-pedagógica, também pode constituir estratégia importante na apresentação e fixação dos tópicos considerados “clássicos”. Da mesma forma, a incorporação dos resultados de alguma pesquisa recente, pode, ainda que em caráter introdutório, constituir elemento de motivação para o estudo, servindo também para promover uma maior e melhor contextualização do conhecimento “clássico” com as pesquisas de ponta na área.

Finalmente, o uso de ferramentas de experimentação prática permite romper a barreira da aridez algébrica, do raciocínio lógico e do pensamento abstrato, permitindo que os alunos com menos facilidade para a manipulação desse tipo de notação e desses mecanismos mentais não seja prejudicado no aprendizado da matéria. O uso de simuladores abertos, em contraposição aos fechados, oferece a vantagem de criar pontos adicionais de contato com outras disciplinas da área de computação, o

que pode contribuir de forma decisiva para a motivação, retenção e aprendizado dos alunos.

As ideias e sugestões apresentadas nesse artigo não esgotam o assunto. Ao contrário, elas são apenas uma tentativa no sentido de revigorar a forma “clássica” de ensino de uma disciplina considerada “clássica”, renovando e aumentando, dessa forma, o interesse dos alunos e dos docentes, e mostrando que, a partir de proposições relativamente simples, é possível construir materiais didáticos e/ou programas de curso diferenciados no que se refere à assimilação do seu conteúdo e à contextualização do mesmo no universo da computação moderna.

As proposições apresentadas no presente artigo são resultado da experiência de vários anos do autor com o ensino da disciplina linguagens formais e autômatos em cursos superiores de computação. Tal experiência resultou na produção de um livro-texto didático, voltado para alunos de cursos superiores de computação, que incorpora os princípios e as ideias aqui discutidas [3].

Referências

- [1] Sociedade Brasileira de Computação, “Currículo de Referência da SBC para Cursos de Graduação em Computação e Informática,” 1999. Disponível em <http://www.sbc.org.br/index.php?language=1&subject=28&content=downloads&id=82>.
- [2] A. N. Chomsky, “Three models for the description of language,” *Information Theory, IRE Transactions on*, vol. 2, no. 3, pp. 113–124, 1956.
- [3] M. V. M. Ramos, J. J. Neto, and I. S. Vega, *Linguagens formais: teoria, modelagem e implementação*. Bookman, 2009. ISBN: 9788577804535.
- [4] A. Salomaa, *Formal languages*. Academic Press Professional, Inc. San Diego, CA, USA, 1987.
- [5] J. J. Neto and M. E. S. Magalhães, “Reconhecedores sintáticos-uma alternativa didática para uso em cursos de engenharia,” in *XIV Congresso Nacional de Informática*, pp. 140–148, 1981.
- [6] J. J. Neto, “Solving complex problems efficiently with adaptative automata,” in *Conference on the Implementation and Application of Automata - CIAA 2000*, CIAA, London, Ontario, Canada, 2000.
- [7] “JFLAP.” Disponível em <http://www.jflap.org/>. Acessado em setembro de 2008.
- [8] “Computing curricula 2001,” *J. Educ. Resour. Comput.*, 2001. Disponível em http://www.computer.org/portal/cms_docs_ieeeecs/ieeeecs/education/cc2001/cc2001.pdf.

- [9] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages and Computation*. Reading, MA: Addison Wesley, 1979.
- [10] T. A. Sudkamp, *Languages and machines: an introduction to the theory of computer science*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 2005.



Mini-currículo do autor: Marcus Vinícius Midena Ramos Engenheiro elétrico (1982) e mestre em sistemas digitais (1990) pela Escola Politécnica da Universidade de São Paulo. Atuou em diversas instituições de ensino superior, entre públicas, privadas e confessionais, em São Paulo e em Recife, em cursos de Bacharelado em Ciência da Computação, Engenharia da Computação e outros relacionados. Sua área de atuação abrange as disciplinas de Linguagens Formais e Autômatos, Teoria da Computação, Compiladores e Linguagens de Programação. Atualmente é professor assistente do colegiado de Engenharia da Computação da Universidade Federal do Vale do São Francisco, sediada em Petrolina, Pernambuco.