

Recursos Computacionais para Suporte ao Ensino de Teoria da Computação, Linguagens Formais e Autômatos

Ícaro Andrade Souza¹, Ecivaldo de Souza Matos¹, Débora Abdalla Santos¹,
Ranansamir Sousa¹

¹Departamento de Ciência da Computação – Universidade Federal da Bahia (UFBA)
40170-110 – Salvador – BA – Brazil

{icaro.andrade; ecivaldo; abdalla; ranansamir.silva}@ufba.br

Abstract. *According to the national curriculum guidelines, Theory of Computation (TC), Automata and Formal Languages (AFL), form a set of basic content to undergraduate courses of Computer Science. Students need to work cognitive aspects such as abstraction and theoretical and mathematical knowledge to learn that content. Over the years, it has been developed several computational resources to support teaching and learning TC and AFL. In this sense, this article presents a brief survey of software to support teaching strategies in Theory of Computation, Automata and Formal Languages, and an analysis of its mainly features and power of simulation.*

Resumo. *Teoria da Computação, Linguagens Formais e Autômatos, segundo as diretrizes curriculares nacionais (DCN), formam um conjunto de conteúdos fundamentais aos estudantes de cursos de graduação em Computação. Para aprendê-los, esses estudantes precisam trabalhar aspectos cognitivos como abstração e conhecimento teórico-matemático. Ao longo dos anos, foram desenvolvidos diversos recursos computacionais destinados a auxiliar o ensino desses conteúdos. Nesse sentido, este artigo apresenta um levantamento de softwares para o enriquecimento das estratégias de ensino de Teoria da Computação, Linguagens Formais e Autômatos, apresentando uma análise do desenvolvimento histórico e de suas funcionalidades para favorecimento da escolha/adoção desses recursos.*

1. Introdução

No Brasil os currículos de cursos superiores são normatizados pelas diretrizes curriculares nacionais (DCN), sob responsabilidade do Ministério da Educação (MEC) (Brasil, 1999). Além das DCN, há parâmetros que norteiam a implantação e revisão curriculares, como os Currículos de Referência da Sociedade Brasileira de Computação (SBC). Segundo as DCN em vigor, para todos os cursos de graduação em Computação (bacharelado/licenciatura), entre os conteúdos curriculares exigidos na formação tecnológica e básica estão Teoria da Computação (TC), Linguagens Formais e Autômatos (LFA).

Alguns autores relatam que os conteúdos curriculares referentes à Teoria da Computação, Linguagens Formais e Autômatos referem-se aos fundamentos matemáticos da Computação, os quais analisam problemas que podem ser computados por um dado modelo teórico de Computação e, de uma forma geral, respondem quais são as capacidades

e as limitações dos computadores (Hopcroft J. E., Motwani, R., Ullman, 2006; Sipser, 2005). Sipser (*ibid*) relata que esses conteúdos podem ser divididos em três partes, oriundas de três grandes teorias: Teoria das Linguagens Formais e dos Autômatos; Teoria da Computabilidade; e Teoria da Complexidade.

Pesquisadores descrevem a importância dos conteúdos curriculares referentes à TC e LFA para os cursos de Computação, enfatizando o quão essenciais são para o desenvolvimento dos estudantes nos cursos de Computação, mais especificadamente para o curso de Ciência da Computação (Chakraborty *et al.*, 2011; Dognini *et al.*, 2003; Sigman 2007). Esses pesquisadores relatam ainda que conteúdos curriculares existentes em TC e LFA dão suporte a outras disciplinas, também fundamentais aos cursos de Computação, a exemplo de Compiladores e Inteligência Artificial.

Desde meados do século passado os conteúdos dessa área já começaram a estabelecer-se como uma importante fundamentação da Ciência da Computação (Chakraborty, 2012). Entretanto, segundo Chakraborty, professores e pesquisadores já percebiam que a TC e LFA possuem temas difíceis de aprender e de ensinar.

Nesse sentido, Ezer e Trakhtenbrot (2005) sinalizam que uma das habilidades-chave necessária aos estudantes de disciplinas referentes aos tópicos de TC e LFA é aptidão para raciocínio matemático preciso, mas isso tem sido difícil e até mesmo assustador para os estudantes. Eles ainda mostram que os problemas ocorridos na tentativa de empregar métodos formais de raciocínio levam a vários equívocos e, por fim, a um conhecimento inadequado. Vijayalaskhmi e Karibasappa (2012) complementam ressaltando que um dos grandes desafios vivenciado por professores ao lecionar disciplinas dessas áreas é devido à natureza matemática do tema e à dificuldade dos estudantes em entender os conceitos de linguagem e teoria de autômatos.

Pirovani e Mataveli (2014) argumentam que os estudantes consideram os tópicos dessas disciplinas áridos, abstratos, complexos e desvinculados de suas atividades profissionais, o que contribui para reduzir o interesse e a motivação pelo seu aprendizado. Borges (2000) ressalta que o modo tradicional de ensino não consegue motivar os estudantes a se interessarem pela disciplina, entre outras razões, pois para esses estudantes não é clara a importância de certos conteúdos para a sua formação.

Outros pesquisadores, como Aguiar e Oeiras (2010), chamam atenção para os livros da disciplina, os quais afirmam que, em sua maioria, são direcionados a conceitos teóricos e matemáticos que acabam por distanciar os estudantes de uma aprendizagem por meio da prática. Os autores afirmam, ainda, que ocorre “baixo uso de ferramentas computacionais de visualização destinadas a superar dificuldades de aprendizagem em Teoria da Computação” (p.818), fato que, segundo eles, também pode contribuir para as dificuldades apresentadas por estudantes e professores.

Pirovani e Mataveli (2014) declaram ainda que “professores têm dificuldades de encontrar formas alternativas para ensinar o conteúdo e tornar a disciplina de Teoria da Computação mais interessante para os alunos” (p.53). O que, de acordo com esses autores, pode ser considerado como um ponto-chave ao desencadeamento de parte dos problemas existentes no ensino dos conteúdos de TC.

Tendo em vista os problemas relacionados ao ensino de conteúdos curriculares referentes à Teoria da Computação, Linguagens Formais e Autômatos, a pesquisa parcialmente relatada neste artigo visa identificar e analisar (i) recursos computacionais e (ii) metodologias educacionais para o ensino de conteúdos curriculares de Teoria da Computação, Linguagens Formais e Autômatos.

Todavia, neste artigo apresentamos os resultados parciais desta pesquisa: identificação e análise de recursos computacionais de apoio ao ensino de Teoria da Computação, Linguagens Formais e Autômatos em cursos de graduação em Computação.

2. Metodologia

Segundo Matos e Silva (2012) “a maioria dos artigos apresentados em conferências renomadas da área, como a conferência internacional do SIGCSE – *ACM Special Interest Group on Computer Science Education*, são reflexões acerca de experiências e introspecções dos seus autores” (p.4). Portanto, torna-se necessário conceber o Ensino de Computação (ou a Educação em Computação) enquanto área de pesquisa, fazendo com que essa nova área possua um diálogo constante e crítico com as ciências da educação (Lister, 2012).

Neste sentido, este artigo relata o processo e os resultados da identificação e análise de alguns recursos computacionais que visam auxiliar o ensino de conteúdos curriculares de Teoria da Computação, Linguagens Formais e Autômatos, cuja investigação foi realizada em duas fases: (i) levantamento de trabalhos sobre recursos computacionais de apoio ao ensino de TC e LFA e (ii) análise dos trabalhos/recursos mapeados. A primeira fase correspondeu a uma pesquisa exploratória com objetivo de mapear os trabalhos relacionados a recursos computacionais referentes a conteúdos curriculares de TC e LFA; a segunda fase tratou da análise dos trabalhos encontrados na primeira fase, buscando identificar recursos computacionais existentes para auxiliar o ensino da área.

Uma pesquisa exploratória foi desenvolvida no sentido de proporcionar uma visão geral acerca de determinado fato, procurando buscar “padrões, ideias ou hipóteses, em vez de testar ou confirmar uma hipótese” (Collis e Hussey, 2005, p. 77).

Na primeira fase da pesquisa, para o desenvolvimento da pesquisa exploratória, foram utilizados os seguintes termos de busca: ensino; Teoria da Computação; Linguagens Formais. Vale destacar que os termos foram buscados em português e em inglês. Foram analisados artigos científicos, trabalhos de dissertação e teses de mestrado/doutorado.

Foram consideradas seis fontes de dados: Biblioteca Digital Brasileira de Teses e Dissertações (BDTD/IBICT); Biblioteca Digital Brasileira de Computação (BDBComp/SBC); Portal de Periódicos da Capes; Web of Science; ACM Digital Library; e a Revista Brasileira de Informática na Educação (RBIE). As bases foram selecionadas por indexarem a maioria dos veículos qualificados na área de Ciência da Computação, bem como pela disponibilidade e acessibilidade. A RBIE foi selecionada por ser o único periódico sobre Informática e Educação brasileiro indexado e sob a chancela da Sociedade Brasileira de Computação (SBC).

Para o desenvolvimento da pesquisa exploratória foram definidos os seguintes critérios de seleção: trabalhos de pesquisa, desenvolvimento tecnológico e/ou relato de

experiência publicados entre 2000 a 2015; escrito em português, inglês ou espanhol; acessíveis gratuitamente por pelo menos uma das fontes selecionadas; apresentar algum recurso computacional que possibilite auxiliar o ensino de tópicos referentes aos conteúdos curriculares de Teoria da Computação, Linguagens Formais e Autômatos.

Visando identificar os recursos computacionais disponíveis para auxiliar no ensino da área, foram analisados 47 trabalhos encontrados referentes a recurso(s) computacional(is) desenvolvidos para auxiliar o ensino de conteúdos relacionados a TC e LFA. Essa análise foi desenvolvida por meio da leitura completa dos trabalhos, identificando cada recurso computacional citado nos textos e identificando as seguintes informações: nome, data, tipos de máquinas que simula, se simula máquinas não determinísticas e se trabalha com outros conteúdos referentes à área. Caso o trabalho não apresentasse todas essas informações, foram realizadas buscas em outros trabalhos relacionados ao recurso com objetivo de completar as informações adquiridas.

3. Resultados

Nesta seção são apresentados os resultados obtidos a partir da primeira e segunda fase do desenvolvimento da pesquisa. Esses resultados referem-se à apresentação e análise dos trabalhos relacionados à área e dos recursos computacionais encontrados que possibilitam auxiliar o ensino de conteúdos curriculares de Teoria da Computação, Linguagens Formais e Autômatos.

3.1. Pesquisa Exploratória

A pesquisa exploratória resultou em 2.965 trabalhos referentes a recursos computacionais e/ou metodologias/métodos de ensino sobre a área. É importante salientar que foram utilizados, em algumas bases de dados, filtros secundários resultando em 2020 trabalhos para analisar. Após a análise dos respectivos títulos, resumos e palavras-chave, 86 artigos foram considerados relevantes para a pesquisa. Entretanto, verificou-se que dentre os 86 artigos selecionados, alguns estavam duplicados, a partir daí foram eliminados os trabalhos ambíguos, restando 63 artigos selecionados. Os dados preliminares dessa pesquisa exploratória podem ser visualizados na Tabela 1.

Tabela 1. Resultados obtidos com a pesquisa exploratória

Base de dados	Total de trabalhos encontrados	Trabalhos selecionados
BDTD (Dissertações)	94	0
BDTD (Teses)	66	0
Porta de Periódicos Capes	666	11
RBIE	16	4
Anais do CSBC/WEI	29	4
BDCOMP	10	10
Web of Science	42	0
ACM	1097	57
Totais	2020	63

Dentre os 63 trabalhos selecionados na pesquisa exploratória, 16 deles se referem a método(s)/metodologia(s) utilizado(s) para o ensino de conteúdo(s) curricular(es) pertencente(s) à Teoria da Computação, Linguagens Formais e Autômatos dos cursos de graduação. Os 47 trabalhos restantes descrevem recurso(s) computacional(is) que possibilita(m) auxiliar o ensino de conteúdo(s) curricular(es) pertencente(s) à área pesquisada.

3.2. Recursos Computacionais

No tocante aos recursos de apoio ao ensino de TC e LFA, foi possível identificar 80 recursos computacionais. Entretanto, para dois desses recursos (*Deus Ex Machina* e *Formal Languages and Automata Environment (FLUTE)*), não foi possível obter informações referente ao poder de simulação ou se simulavam máquinas não-determinismo; portanto, eles foram desconsiderados dos resultados finais da pesquisa, restando 78 recursos.

Para cada um dos 78 recursos computacionais então considerados, foram mapeadas as seguintes informações: *ano do último trabalho publicado referente à descrição do recurso; nome do recurso computacional; os tipos de máquinas reconhecidas pelos recursos (AF - Autômato Finito, AP – Autômato de Pilha e MT – Máquina de Turing); se o recurso computacional simula/trabalha de alguma forma conteúdos referentes a máquinas não-determinísticas; e, por fim, se o recurso apresenta outra(s) funcionalidade(s)/característica(s), dentre ela(s) esta(ão): trabalhar com algum tipo de linguagem ou gramática da hierarquia de Chomsky; bem como algoritmos de simplificação e/ou conversão de gramáticas e/ou conversão de autômatos; além de recursos que reconhecem máquinas de Moore; máquinas de Mealy; ou pumping lemas.*

A partir dessas informações, foi desenvolvido um panorama da criação/desenvolvimento desses recursos ao longo dos anos e gráficos que apresentam a relação dos recursos encontrados com o poder de simulação das máquinas referentes à hierarquia de Chomsky e a outros conteúdos da área. Esses trabalhos permitiram obter mais informações sobre recursos computacionais que possibilitam auxiliar o ensino de TC e LFA, apresentados resumidamente nas próximas subseções.

3.2.1 Panorama da criação/desenvolvimento de Recursos Computacionais

Um levantamento histórico da criação/desenvolvimento de recursos computacionais, destinados ao ensino de conteúdos referentes à TC e LFA, foi desenvolvido a partir dos dados referentes aos 78 recursos computacionais encontrados. Para tanto utilizou-se dos dados referentes ao ano de cada recurso, sendo que o ano considerado pelo levantamento refere-se ao último trabalho encontrado que descreve a criação/modificação do mesmo, uma vez que caso haja outros trabalhos referentes ao recurso, o último possivelmente trata-se da versão mais completa/estável.

O primeiro recurso computacional encontrado (*Simulation of Turing Machine on a Digital Computer*) foi desenvolvido na década de 1960. O panorama do desenvolvimento dos recursos computacionais encontrados até 2015, está apresentado no Gráfico 1.

Observou-se que nas décadas de 1960, 1970 e 1980 o desenvolvimento desses recursos foi pouco significativo com relação às décadas seguintes. Uma vez que, a partir da década de 1990 houve crescimento superior ao dobro dos recursos desenvolvidos

anteriormente. Segundo os dados obtidos na segunda fase da pesquisa, dos recursos encontrados o maior número situa-se entre os anos de 2000 e 2009. Possivelmente esse crescimento esteja relacionado à evolução tecnológica ocorrida na época, bem como ao surgimento/popularização de novos dispositivos eletrônicos, os quais possivelmente favoreceram a comunicação e a manipulação iterativa de informações.

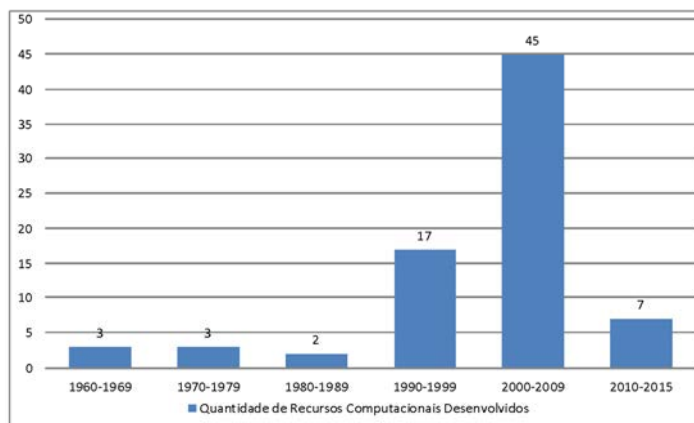


Gráfico 1. Desenvolvimento de recursos computacionais para TC e LFA ao longo dos anos

Conforme os resultados encontrados, a partir da década de 1990 houve crescimento no desenvolvimento desses recursos computacionais destinados a apoiar o ensino de TC e LFA, entretanto nos últimos cinco anos ocorreu considerável redução no número de recursos desenvolvidos.

As informações obtidas que se referem ao desenvolvimento de recursos computacionais entre os anos 2010 e 2015, *a priori* parecem contraditórias ao cenário atual, dado o desenvolvimento contínuo de novas tecnologias e consequente aumento na utilização de recursos tecnológicos em diversas áreas de ensino. Fato que suscitou os seguintes questionamentos: (i) *Por que no período entre 2000 e 2009 houve desenvolvimento desses recursos tão superior aos demais períodos?* (ii) *Por que no período de 2010 a 2015 houve uma queda significativa no desenvolvimento dos recursos computacionais para o ensino de TC e LFA?* Esses questionamentos não foram tratados nesta fase da pesquisa, entretanto poderão ser abordados em fases futuras.

3.2.2 Poder de Simulação dos Recursos Computacionais

Dentre os 78 recursos computacionais, cada um deles possui um tipo de simulação, alguns simulam estritamente autômatos finito (AF), outros autômatos de pilha (AP) ou máquinas de Turing (MT). Todavia, há recursos que simulam mais de um tipo de máquina e, por isso, podem ser utilizados para auxiliar o ensino de mais de um conteúdo curricular da área. Além disso, há recursos que auxiliam o ensino de outros conteúdos da área, como: linguagens, gramáticas, funções de conversão e simplificação, conforme Gráfico 2.

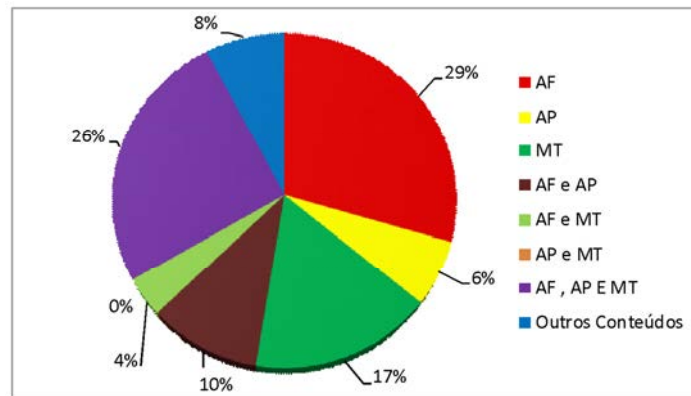


Gráfico 2. Capacidade de simulação dos recursos computacionais encontrados

Nota-se que há mais recursos computacionais destinados a simular os autômatos finitos, possivelmente por se tratar do reconhecedor das Linguagens Regulares que se refere à primeira classe da hierarquia de Chomsky, mais simples e geralmente a primeira a ser apresentada nas disciplinas de graduação relacionados à Teoria da Computação, Linguagens Formais e Autômatos.

Observou-se através dos recursos computacionais encontrados que inicialmente eles eram produzidos para simular uma máquina, seja ela AF, AP ou MT; entretanto, com o passar do tempo foram surgindo recursos que simulavam mais de um tipo de máquina. Segundo os dados obtidos nesta pesquisa temos o *Hypercard Automata Simulation*, no ano de 1992, como primeiro recurso a simular os três tipos de máquinas da hierarquia.

A partir de então é possível encontrar diversos outros recursos com poder de simulação equivalente. Conforme a busca realizada, atualmente há 20 recursos que possuem a capacidade de simular os três tipos de máquinas da hierarquia, totalizando o segundo maior grupo dos recursos encontrados. Possivelmente isso está relacionado ao fato de que quanto mais conteúdos o recurso computacional abranger, mais possibilidades de ele ser utilizado; o que, por outro lado, não pode ser utilizado como parâmetro de qualidade.

O não-determinismo é um conceito fundamental em Ciência da Computação; entretanto, por se tratar de um conceito muito abstrato é difícil de ensinar e aprender (Armoni et al., 2008). As máquinas/autômatos presentes na hierarquia de Chomsky tratam de dois conceitos: determinismo e não determinismo. Isso, segundo Armoni et al., é uma oportunidade para os alunos entenderem o significado e a importância desses conceitos.

Embora o conceito de não-determinismo seja importante para as disciplinas referentes a TC e LFA, diversos recursos computacionais não simulam máquinas não-determinísticas. Os recursos encontrados foram classificados conforme o(s) tipo(s) de máquina(s) que reconhecem (AF, AP e MT) e sua capacidade de simular as máquinas não-determinísticas (Gráfico 3).

descrição, tipos de conteúdos abrangidos, ano da última versão desenvolvida, *link* para acessar e manual para utilização.

4. Considerações Finais

Este artigo apresentou identificação e avaliação de recursos computacionais que possibilitam apoiar o ensino de conteúdos referentes à Teoria da Computação, Linguagens Formais e Autômatos em cursos de graduação em Computação. Dentre os resultados obtidos, destaca-se o resumo cronológico da criação/desenvolvimento dos recursos encontrados, o qual permitiu observar que nos últimos cinco anos houve uma queda significativa no desenvolvimento desses recursos, suscitando os seguintes questionamentos: (i) *Por que no período entre 2000 e 2009 houve desenvolvimento desses recursos tão superior aos demais períodos?* (ii) *Por que no período de 2010 a 2015 houve uma queda significativa no desenvolvimento dos recursos computacionais para o ensino de TC e LFA?*

Outro resultado a se destacar foi à avaliação da capacidade de simulação dos recursos computacionais, a qual constata-se que dos 78 recursos computacionais encontrados, 20 possuem capacidade de simular as máquinas (AF, AP e MT), dentre esses, 14 simulam as máquinas não-determinísticas. E destes, sete podem ser destacados por conter outras funcionalidades referentes a conteúdos da área. Embora os recursos computacionais possam apoiar o ensino, eles não podem ser considerados como substitutos ao professor nem a outros métodos, técnicas ou estratégias de ensino.

Como trabalho futuro, pretende-se aprofundar a análise por meio de experimentos controlados e avaliação da interação tecnológica com uma ou mais propostas metodológicas. Nesse processo serão identificados e analisados métodos/metodologias para ensino de TC e LFA. Com isso espera-se ao final desta pesquisa apresentar possibilidades metodológicas para o ensino de conteúdos curriculares de Teoria da Computação, Linguagens Formais e Autômatos.

5. Referências

- Aguiar, S. and Oeiras, J. Y. Y. (2010). Ambiente Moodle de auxílio ao ensino e aprendizagem em Linguagens Formais. XVIII Workshop sobre Educação em Informática. XXX Congresso da Sociedade Brasileira de Computação. p. 818–827.
- Armoni, M., Lewenstein, N. and Ari, M. Ben (2008). Teaching students to think nondeterministically. ACM SIGCSE Bulletin, p. 4–8.
- Borges, M. (2000). Avaliação de uma metodologia alternativa para a aprendizagem de programação. VIII Workshop de Educação em Computação (WEI). XX Congresso da Sociedade Brasileira de Computação.
- Brasil (1999). Diretrizes Curriculares Nacionais para os cursos de graduação em Computação. Disponível em: http://portal.mec.gov.br/index.php?option=com_content&id=12991:diretrizes-urriculares-cursos-de-graduacao. Acessado em: 20 de abril de 2016.
- Chakraborty, P., Saxena, P. C. and Katti, C. P. (2011). Fifty years of automata simulation: a review. ACM Inroads, v. 2, n. 4, p. 59–70.

- Chakraborty, P., Saxena, P. C. and Katti, C. P. (2012). Automata simulators: Classic tools for computer science education. *British Journal of Educational Technology*, v. 43, n. 1, p. 2011–2013.
- Chesñevar, C. I., González, M. P. and Maguitman, A. G. (2004). Didactic strategies for promoting significant learning in formal languages and automata theory. *ACM SIGCSE Bulletin*, v. 36, n. 3, p. 7–11.
- Collis, J., Hussey, R. (2005). *Pesquisa em administração: um guia prático para alunos de graduação e pós-graduação*. Porto Alegre: Bookman, p. 352.
- Dognini, M. J., Luís, A. and Raabe, A. (2003). EduLing - Software Educacional para Linguagens Regulares. XIV Simpósio Brasileiro de Informática na Educação – NCE – IM/UFRJ,
- Ezer, J. G. and Trakhtenbrot, M. (2005). Challenges in teaching the pumping lemma in automata theory course. *ACM SIGCSE Bulletin*, v. 37, n. 3, p. 369.
- Hopcroft J. E., Motwani, R., Ullman, J. D. (2006). *Introduction to Automata Theory, Languages and Computation*. International Edition.
- Lister, R. (2012). Teaching-oriented faculty and computing education research. *ACM Inroads*, v. 3, n. 1, p. 22–23.
- Matos, E. and Silva, G. Da (2012). Currículo de licenciatura em computação: uma reflexão sobre perfil de formação à luz dos referenciais curriculares da SBC. *Anais do XXXII Congresso da Sociedade Brasileira de Computação - XX Workshop sobre Educação em Computação (WEI)*.
- Pirovani, J. C. and Mataveli, G. V. (2014). Estudo e adaptação de software para o ensino de Linguagens Formais e Autômatos. *Revista Brasileira de Informática na Educação*, v. 21, p. 53–68.
- Schez, J. J. C., Castillo, E. Del, Hortolano, J. and Rodriguez, A. (2009). Designing and using software tools for educational purposes: FLAT, a case study. *IEEE Transactions on Education*, v. 52, n. 1, p. 66–74.
- Sigman, S. (2007). Engaging Students in Formal Language Theory and Theory of Computation. *SIGCSE '07: Proceedings of the 38th SIGCSE technical symposium on Computer science education*, p. 450–453.
- Sipser, M. (2005). *Introduction to the Theory of Computation*. Second Edition.
- Tarouco, L. and Berch, M. (2009). O uso de mobile learning no ensino de algoritmos. *Novas Tecnologias na Educação*, v. 7, n. 3, p. 4.
- Vijayalaskhmi, M. and Karibasappa, K. (2012). Activity based teaching learning in formal languages and automata theory-An experience. *Engineering Education: Innovative Practices and Future Trends (AICERA), 2012 IEEE International Conference on*