

Estudo e adaptação de *software* para o ensino de Linguagens Formais e Autômatos

Title: Study and adaptation of software for teaching Formal Languages and Automata

Juliana Pinheiro Campos Pirovani

Universidade Federal do Espírito Santo - CCA Alto Universitário, s/nº - Depto. Computação Cx Postal 16, Guararema – 29500-000 Alegre-ES jupcampos@gmail.com

Guilherme Vezula Mataveli

Universidade Federal do Espírito Santo – CCA Alto Universitário, s/nº - Depto. Computação Cx Postal 16, Guararema – 29500-000 Alegre-ES guillhermemataveli@hotmail.com

Resumo

A disciplina Linguagens Formais e Autômatos é fundamental nos cursos superiores de computação por estudar características das linguagens de programação (que são linguagens formais), os modelos computacionais que reconhecem essas linguagens e outros modelos que permitem a especificação dessas linguagens. O alto nível de abstração exigido nessa disciplina contribui para que os alunos apresentem dificuldades no seu aprendizado. Por outro lado, os professores têm dificuldades de encontrar formas alternativas para ensinar o conteúdo e tornar a disciplina mais interessante para os alunos. Este artigo apresenta algumas adaptações e implementação de novas funcionalidades realizadas em uma ferramenta computacional (software) com o objetivo de auxiliar o ensino e aprendizagem de Linguagens Formais e Autômatos. Para isso, foi realizada uma análise das ferramentas computacionais existentes para o ensino dessa disciplina, onde a ferramenta JFLAP foi selecionada para a inclusão de novas funcionalidades. Também foi realizado um estudo de caso com o objetivo de identificar a necessidade de adaptações na ferramenta. E, por último, a adaptação do software foi realizada. Acredita-se que essa pesquisa possa contribuir para melhoria do ensino da disciplina Linguagens Formais e Autômatos em cursos superiores de computação e ampliar a utilização da ferramenta no processo de ensino e aprendizagem.

Palavras-Chave: linguagens formais, autômatos, ferramenta de ensino.

Abstract

The discipline Formal Languages and Automata is fundamental in courses of higher education computing by studying characteristics of programming languages (which are formal languages), computational models that recognize these languages and other models that allow the specification of these languages. The high level of abstraction required in this discipline increases the difficulties that students have in their learning. Moreover, teachers have difficulties to find alternative ways to teach content and make the discipline more interesting for students. This article presents some adjustments and implementation of new functionalities performed in a computational tool (software) in order to assist teachers in teaching Formal Languages and Automata. So, we conducted an analysis of existing computational tools for the teaching of this discipline, where the JFLAP tool was selected for the inclusion of new features. There was also a case study in order to identify the need for adjustments in the chosen tool. And last, the adaptation of the software was performed. It is believed that this research will contribute to improving the teaching of the subject in Automata and Formal Languages courses in computing and expand the use of the tool in the process of teaching and learning.

Keywords: formal languages, automata, teaching tool.

Recebido: 25 de Junho de 2013 / Aceito: 28 Dezembro de 2013

DOI: 10.5753/RBIE.2013.21.03.53



1 Introdução

A teoria de Linguagens Formais e Autômatos (LFA) estuda as classes de linguagens que possuem sintaxe bem definida e semântica precisa [1], bem como os modelos computacionais que reconhecem essas linguagens e que permitem sua especificação.

O estudo de disciplinas teóricas com definições que utilizam formalismos matemáticos e demonstrações de teoremas, como acontece em LFA, pode torná-las complexas e pouco interessantes para os alunos. O estudo dessa disciplina contempla tópicos considerados pelos alunos excessivamente áridos, abstratos, complexos e desvinculados da sua futura atividade profissional, o que contribui para reduzir o interesse e a motivação pelo seu aprendizado [2].

Sendo assim, os professores possuem a árdua tarefa de buscar formas alternativas para o ensino do conteúdo com o objetivo de tornar a disciplina mais atrativa para os alunos.

Os professores devem apresentar o conteúdo de forma clara, traduzindo toda linguagem matemática em termos mais coloquiais. Devem também apresentar as aplicações práticas de todo conteúdo para que o aluno compreenda o quanto a teoria estudada é importante. Segundo Sipser [3], aprender qualquer assunto fica mais fácil e divertido se o assunto é devidamente apresentado.

De acordo com Oliveira [4], as ferramentas computacionais utilizadas como suporte ao processo de ensino-aprendizagem, rendem largas oportunidades para a construção crítica do conhecimento. Elas podem oportunizar, no contexto acanhado da sala de aula e para além dele, a dinâmica da experimentação.

Atualmente existem várias ferramentas que podem auxiliar o ensino e aprendizagem de Linguagens Formais e Autômatos [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]. Essas ferramentas são capazes de facilitar o entendimento dos alunos ao possibilitar o projeto de modelos computacionais e a execução de testes. Elas proporcionam uma aprendizagem bastante eficiente para os alunos, permitindo a construção de modelos durante a realização de exercícios e possibilitando a verificação dos mesmos. Apesar disso, observou-se em uma análise preliminar que alguns recursos e funcionalidades poderiam ser incluídos em ferramentas desse tipo para auxiliar o professor no ensino de alguns conceitos e os alunos ao estudar o conteúdo.

Este trabalho foi desenvolvido com o objetivo de adaptar uma ferramenta computacional, implementando novas funcionalidades nessa ferramenta que possam auxiliar o ensino e aprendizagem de Linguagens Formais e Autômatos.

A seção 2 apresenta algumas ferramentas de auxilio ao ensino de LFA que foram estudadas com o objetivo de analisar as funcionalidades incluídas em cada uma e selecionar uma delas para posterior adaptação.

A seção 3 apresenta um estudo de caso realizado com o objetivo de avaliar a utilização da ferramenta escolhida e tentar identificar a necessidade de adicionar outras funcionalidades nessa ferramenta. Na seção 4 são apresentadas algumas modificações e adaptações realizadas na ferramenta escolhida. Na seção 5 é apresentado outro estudo de caso, realizado para validação do trabalho. E, na seção 6, as conclusões.

Os documentos utilizados nos estudos de caso são apresentados ao final do documento, como apêndices.

2 Ferramentas que Auxiliam o Ensino de LFA

Os *softwares* que auxiliam o ensino e aprendizagem de Linguagens Formais e Autômatos investigados e analisados foram: GAM [5], VAS [6], Auger [7, 8], Language Emulator [9, 10], SCTMF [11, 12], JFLAP [13, 14, 15] e LabLF [16].

Todas essas ferramentas permitem a criação e simulação de modelos computacionais e podem ser utilizadas por alunos e professores em diversos contextos. De acordo com Costa et al. [12], a SCTMF [11, 12] pode ser utilizada nas situações apresentadas a seguir, onde também poderiam ser utilizadas qualquer uma das ferramentas analisadas neste trabalho:

- auxiliar didaticamente o professor na exemplificação de modelos formais e demonstração da dinâmica de funcionamento dos mesmos em sala de aula;
- auxiliar os alunos na construção de modelos formais em aulas práticas em laboratório;
- auxiliar os alunos na solução de exercícios de forma não presencial.

Além dessas situações, essas ferramentas também podem ser utilizadas para auxiliar o professor na correção de exercícios e os alunos na verificação dos exercícios propostos pelo professor.

O que diferencia as ferramentas estudadas são os tipos de modelos computacionais que podem ser construídos e simulados em cada uma e a forma de entrada de dados ao criar os modelos. Algumas ferramentas permitem que o diagrama de estados do modelo seja construído e manipulado por meio de um editor gráfico, inserindo e removendo círculos e setas, representando os estados e transições respectivamente. A Figura 1 mostra esse tipo de entrada de dados na ferramenta Auger [7, 8]. As ferramentas GAM [5], VAS [6], JFLAP [13, 14, 15] e LabLF [16] também



possuem esse formato de entrada para criação dos modelos.

Outras ferramentas não possibilitam essa edição visual do diagrama de estados, como a Language Emulator [9, 10] e a SCTMF [11, 12]. Nesse caso, a entrada de dados é realizada inserindo um a um os componentes da definição formal do modelo. Por exemplo, um autômato finito determinístico (AFD), consiste formalmente em uma 5-upla (S, Σ , δ , q0, F) onde [3]:

- S é um conjunto finito de estados.
- Σ é um conjunto finito de símbolos de entrada, o alfabeto.
- δ: S x Σ → Q representa uma função de transição que toma como argumentos um estado e um símbolo de entrada e retorna um estado. Se q é um estado e a um símbolo de entrada, então δ(q,a) = p indica que existe uma seta identificada por a de q até p.
- $q0 \in S$ é o estado inicial.
- F é um subconjunto de S representando os estados finais do autômato.

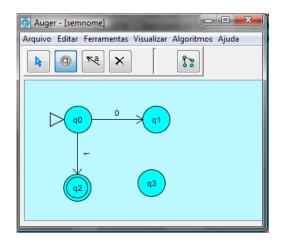


Figura 1. Entrada de dados para criar Autômato Finito Determinístico no Auger [7, 8].

Para inserir os dados de um AFD no SCTMF [11,12] é necessário preencher as informações de cada componente dessa definição. A Figura 2 apresenta um exemplo de inserção da função de transição nessa ferramenta:

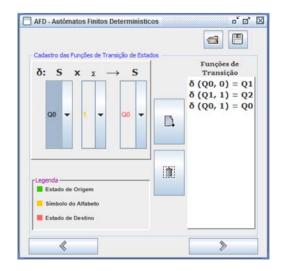


Figura 2. Entrada de dados para criar AFD no SCTMF [11,12].

A Tabela 1 apresenta uma comparação entre as ferramentas analisadas. Na primeira coluna são apresentados alguns modelos computacionais estudados em LFA e algumas possibilidades de conversão entre os modelos. São eles: Autômato Finito Determinístico (AFD), Autômato Finito Não Determinístico (AFN), Autômato Finito Não Determinístico com transição vazia (AFN-ε), Gramática Regular (GR), Expressão Regular (ER), Máquina de Moore (M. Moore), Máquina de Mealy (M. Mealy), Autômato de Pilha (AP), Máquina de Turing (MT), conversão de AFN para AFD (AFN → AFD), minimização de AFD (Min. AFD), conversão de Autômato Finito para Expressão Regular (AF → ER), conversão de ER para AF (ER → AF), conversão de AFD para GR (AFD → GR), conversão de Gramática Livre de Contexto para AP (GLC \rightarrow AP). As outras colunas apresentam o que está implementado em cada ferramenta (marca com X as características que a ferramenta possui).

As ferramentas Ginux Abstract Machine (GAM) [5] e Visual Automata Simulator (VAS) [6] são as mais simples das ferramentas investigadas. GAM permite a criação e simulação de AFD, AFN, AFN-ɛ e as conversões AFN -> AFD e Min.AFD. Além desses modelos, a VAS possibilita a criação de MT, mas não permite minimizar um AFD. Ela possui código aberto, o que possibilita modificar o código e inserir outras funcionalidades.

A ferramenta Auger [7, 8] foi desenvolvida com o objetivo de apoiar o ensino de autômatos finitos. Como é possível observar na tabela 1, suas funcionalidades possibilitam apenas a construção desses tipos de autômatos e transformações sobre eles. O diferencial dessa ferramenta é a capacidade de gerar um programa reconhecedor em Java para os autômatos construídos.

Language Emulator [9, 10] é uma ferramenta que permite, além da construção de autômatos finitos, a construção de GR, ER, M. Mealy e M. Moore, mas não



implementa AP e MT (um modelo bastante utilizado nas disciplinas de LFA). Além disso, a forma de entrada dos dados é pouco intuitiva, já que não permite a edição visual do diagrama de estados. De acordo com Vieira et al. [10], o uso da ferramenta obteve uma aceitação de 95% em pesquisa realizada com os alunos.

Já o Software de Criação e Teste de Modelos Formais (SCTMF) [11, 12] permite a construção de AP e MT, além dos modelos relacionados a autômatos finitos que a Language Emulator permite. Porém, não permite algumas conversões importantes, como AF -> ER, ER -> AF. Essa ferramenta possui a mesma desvantagem da Language Emulator: a entrada de dados é feita formalmente. Avaliações realizadas com essa ferramenta sugerem que quanto maior a complexidade do modelo a ser construído, maior tende a ser o auxílio prestado pela ferramenta. Isso porque os testes de verificação contribuem mais em termos de auxiliar na identificação de erros cometidos na criação do modelo [12].

Como é possível observar na Tabela 1, a ferramenta Java Formal Languages and Automata Package (JFLAP) [13, 14, 15] é a mais completa das que foram estudadas. Ela foi desenvolvida por pesquisadores norte-americanos e é capaz de representar grande parte dos modelos computacionais existentes (AFD, AFN, AP, MT, MT com múltiplas fitas e outros). Após a criação dos modelos, os usuários podem testá-los (simular a computação para algumas palavras de entrada), como nas outras ferramentas. Porém, o JFLAP possibilita visualizar a computação de uma palavra de entrada passo-a-passo no modelo antes de informar se a palavra foi aceita ou não. Isso facilita o acompanhamento da simulação pelos alunos e o ensino do funcionamento do autômato pelo professor. Além disso, o JFLAP possibilita também a conversão entre vários modelos computacionais equivalentes.

Segundo Chakraborty et al. [15], o JFLAP é a ferramenta mais sofisticada, bem documentada e utilizada entre as ferramentas para simular autômatos.

Já a ferramenta chamada Laboratório de Linguagens Formais (LabLF) [16] é uma ferramenta que foi criada com objetivo um pouco diferente das já apresentadas. Além de criar e simular modelos computacionais, ela foi construída a fim de permitir a elaboração de listas de exercícios ou avaliações que podem ser corrigidas de modo automático ou semi-automático. Para fazer isso, os editores do JFLAP foram reutilizados nessa ferramenta possibilitando a construção dos modelos e foram

Características	GAM	VAS	Auger	Language Emulator	SCTMF	JFLAP	LABLF
AFD	Χ	Χ	Х	Χ	Х	Χ	Х
AFN	Χ	Χ	Х	Χ	Х	Χ	Х
AFN-ε	Χ	X	Х	Χ	Х	Χ	Х
GR				Χ	Х	Х	Х
ER				Х	Х	Χ	Х
M. Moore				Χ	Х	Х	
M. Mealy				Х	Х	Χ	
AP					Х	Х	Х
MT		Χ			X	Χ	Χ
AFN -> AFD	Х	Х	Х	Χ	Х	Х	
Min. AFD	Χ		Х	Χ	X	Χ	
AF -> ER			Х	Χ		Χ	
ER -> AF			Х	Χ		Χ	
AFD -> GR			Х	Х		Χ	
GLC -> AP						Χ	

Tabela 1: Ferramentas computacionais analisadas e suas características.

implementados recursos para que ela pudesse fazer correções imediatamente e dizer aos alunos se suas soluções estavam ou não corretas. Ou seja, LabLF fornece feedback rápido sobre a solução de exercícios propostos pelo professor aos alunos.

A proposta deste trabalho foi adaptar uma dessas ferramentas computacionais, inserindo funcionalidades que ainda não haviam sido implementadas em nenhuma delas e que pudessem auxiliar o ensino e aprendizagem de LFA.

Para atender a proposta deste trabalho, foi necessário

escolher uma dessas ferramentas estudadas para posterior adaptação. A ferramenta escolhida para incluir as novas funcionalidades deveria ser um *software* livre. Como o *software* livre possui código-fonte aberto, o *software* escolhido poderia ser estudado para implementação das novas funcionalidades.

Apenas duas das ferramentas estudadas possuem código fonte aberto: VAS [6] e JFLAP [13, 14, 15]. O JFLAP foi o escolhido para adaptação neste trabalho por possuir uma quantidade maior de funções já desenvolvidas e também por apresentar a computação de uma entrada passo-a-passo nos modelos, o que auxilia no



acompanhamento da simulação. O JFLAP foi desenvolvido em JAVA e pode ser executado em qualquer computador que tenha a *Java Virtual Machine* (JVM) instalada.

3 Estudo de Caso

Para o estudo de caso foi elaborado um questionário (ver Apêndice A). Esse questionário foi aplicado aos 5 alunos que cursavam a disciplina Teoria da Computação, oferecida no 5º período do Bacharelado em Ciência da Computação do Centro de Ciências Agrárias da Universidade Federal do Espírito Santo – CCA/ UFES, e para os 12 alunos que cursavam a disciplina Computabilidade e Complexidade, oferecida no 3º período do Bacharelado em Sistemas de Informação do CCA / UFES no semestre 2011/2. Essas duas disciplinas apresentam o conteúdo de Linguagens Formais e Autômatos. Os alunos de Teoria da Computação já haviam utilizado o JFLAP no semestre anterior e os de Computabilidade e Complexidade o utilizavam pela primeira vez.

Analisando os questionários respondidos pelos alunos no estudo de caso, foi observado que os participantes que utilizam o JFLAP por menos de 6 meses (65% dos alunos) se consideram usuários iniciantes do *software*. Já aqueles que utilizam o JFLAP de 6 meses a 1 ano se consideram usuários intermediários. Isso indica que em um tempo relativamente curto (até 1 ano), os usuários conseguem utilizar as funcionalidades principais do JFLAP, ou seja, criar e simular os principais modelos computacionais.

Apesar disso, os alunos relataram que sentiram dificuldades ao utilizar o JFLAP pela primeira vez. 71% dos participantes disseram que conseguiram utilizá-lo apenas após a ajuda de um colega ou professor, conforme apresentado na Figura 3.



Figura 3: Gráfico com a opinião dos alunos sobre a utilização da ferramenta JFLAP [13].

Com relação à documentação oferecida pelo JFLAP, 65% dos alunos participantes da pesquisa nunca recorreram a documentação. Apenas 2 alunos (que corresponde a 12% dos participantes) disseram que a documentação poucas vezes foi suficiente para sanar as

dúvidas que surgiram durante o uso do software.

Os alunos reconhecem a importância da utilização do *software* para seu aprendizado: 71% dos alunos disseram que o JFLAP foi importante no seu aprendizado e 23% consideraram muito importante (Figura 4). Esse resultado confirma a importância da utilização de ferramentas computacionais como suporte ao processo de ensino-aprendizagem. No caso do JFLAP, o seu uso em sala de aula permite aos alunos compreender mais facilmente os tópicos estudados através da experimentação, da criação e simulação de modelos computacionais. Isso aumenta o interesse e a motivação dos alunos pela disciplina.



Figura 4: Gráfico com a opinião dos alunos sobre a importância da ferramenta JFLAP [13].

Dos 17 participantes do estudo de caso, 47% disseram que a interface em inglês não prejudica a compreensão do programa e 29% respondeu que poucas vezes prejudica, como apresenta a Figura 5. Apesar disso, ao final do questionário os participantes sugeriram modificações e a maioria dos alunos que se consideram iniciantes (63%) sugeriram a tradução da interface do software.



Figura 5: Gráfico com a opinião dos alunos sobre a interface da ferramenta JFLAP [13].

Alguns alunos da turma de Computabilidade e Complexidade (18%) relataram comportamentos inesperados ocorridos durante a utilização do JFLAP que apresentaram possibilidades de modificação do *software*. Por exemplo, alguns alunos relataram problemas ocorridos ao tentar gerar um autômato a partir de uma

Revista Brasileira de Informática na Educação, Volume 21, Número 3, 2013



expressão regular. Observou-se que essa funcionalidade apresentava erros de lógica, incluindo vários estados e transições desnecessárias no autômato.

Outro problema relatado pelos alunos ocorreu durante a simulação de múltiplas entradas numa máquina de Turing transdutora (máquina com entrada e saída). Após computar a função, a saída não era apresentada pois o cabeçote da máquina não era posicionado no primeiro símbolo da cadeia.

No final do questionário os alunos tinham a liberdade de sugerir melhorias que julgavam importantes para seu aprendizado. Algumas delas são descritas abaixo:

- Possibilidade de copiar e colar um modelo ou parte dele.
- Adicionar teclas de atalho como Delete e Insert (para deletar e inserir partes do modelo respectivamente).
- Mover uma transição já existente arrastando a linha de transição.
- Possibilidade de abrir um modelo a partir do arquivo jff salvo.

Foi observado que algumas sugestões foram feitas por falta de conhecimento ou entendimento do *software* e acredita-se que poderiam ser resolvidas com a sua tradução para o português. Por exemplo, os alunos sugeriram que fosse possível copiar um modelo ou parte dele. Existe uma função no *software* que permite combinar autômatos (Convert → Combine automata), abrindo os autômatos solicitados em uma nova janela. Essa função permite utilizar modelos prontos.

Foi observado também que algumas das dificuldades apresentadas pelos alunos ao usar o JFLAP poderiam ser sanadas ou amenizadas com modificações simples sugeridas pelos participantes do estudo de caso e que poderiam melhorar a usabilidade do software.

A partir disso, adaptações e novas funcionalidades que seriam implementadas foram definidas e priorizadas.

4 Correções e Implementações na Ferramenta de Apoio a LFA

Inicialmente foram incluídas as funcionalidades cuja necessidade foi detectada pela professora das disciplinas da área de LFA durante o uso do *software* em suas aulas. Posteriormente, foram implementadas outras funcionalidades necessárias identificadas durante o estudo de caso.

Um dos principais problemas desenvolvedores é implementar novas funcionalidades e gerir mudanças em sistemas desenvolvidos por terceiros. Essa foi uma das principais dificuldades encontradas durante a realização desse projeto. Vários programadores contribuíram no desenvolvimento do JFLAP e por isso não existe um padrão na sua documentação e programação. Isso dificultou o entendimento do mesmo e, consequentemente, impossibilitou que outras modificações e/ou correções fossem realizadas imediato. Elas serão implementadas posteriormente.

Para cada adaptação proposta foram realizados os seguintes passos:

- Planejamento de como a adaptação seria realizada no software: identificação do local onde seria visualizada a adaptação ou acionada uma nova funcionalidade na interface.
- Análise do código fonte: identificação do local onde seria inserida a adaptação e como ela seria acrescentada no código fonte. Exigiu um estudo anterior para compreensão do programa e da linguagem Java.
- Análise de impacto da alteração: verificação e análise dos possíveis efeitos indesejados inseridos em outras funcionalidades do JFLAP devido a alteração.
- Modificação do código fonte: Implementação da adaptação no software.
- Documentação da modificação do código fonte: documentação explicitando detalhes da implementação da modificação e outros possíveis locais onde ela se aplica.
- Testes: execução de vários testes para verificar o correto funcionamento da adaptação inserida.

A estrutura fundamental do *software* permaneceu estável. As adaptações foram feitas em locais específicos para atender as necessidades identificadas.

A primeira modificação realizada no JFLAP foi a inclusão de uma legenda no painel de simulação de autômatos finitos. Ela foi realizada porque alguns alunos relataram que não tinha como identificar se a computação da *string* testada estava em processamento, se já tinha finalizado e a *string* tinha sido aceita ou rejeitada. Deste modo, foi adicionada a referida legenda com cores no painel de simulação, identificando o estado da simulação como apresentado na Figura 6.



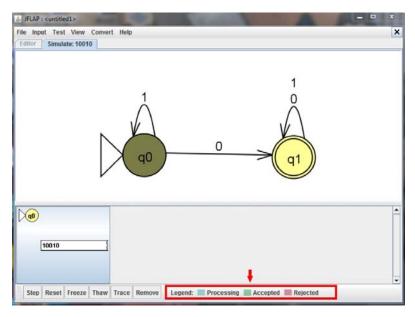


Figura 6: Painel de simulação de autômatos com o acréscimo da legenda.

Duas operações que podem ser aplicadas a um autômato finito foram implementadas: a operação estrela (fecho de Kleene) e o complemento de um autômato. As duas operações podem ser aplicadas a qualquer AFD ou AFN que tenha estado inicial e final. A Figura 7

apresenta um AFD no JFLAP e a Figura 8 apresenta a aplicação da operação estrela (a) e a aplicação da operação complemento (b) no AFD da Figura 7. A Figura 8 destaca os novos recursos na interface.

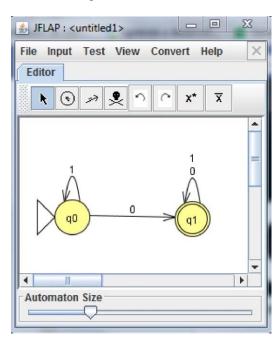


Figura 7: Exemplo de AFD no JFLAP [13].



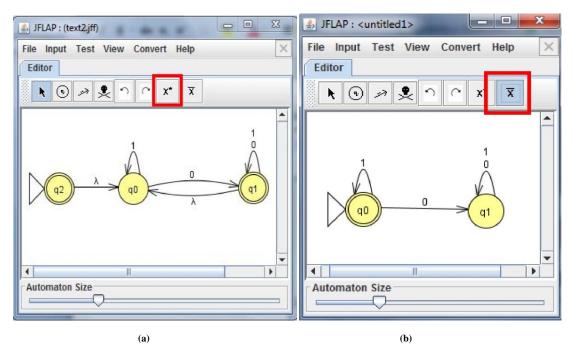


Figura 8: Exemplo da implementação de funcionalidades no painel de edição do JFLAP [13] onde: (a) apresenta a aplicação da operação estrela; e (b) apresenta a aplicação da operação complemento sob o autômato apresentado na Figura 5.

Outras operações adicionadas ao JFLAP foram a união e concatenação de dois autômatos. Para realizar essas operações basta criar dois autômatos ou abrir dois autômatos salvos. Dois botões foram adicionados ao painel de edição e eles se tornam visíveis apenas quando

existem dois autômatos abertos no *software*. A Figura 9 apresenta dois AFDs no JFLAP e a Figura 10 apresenta a aplicação da operação união (a) e a aplicação da operação concatenação (b) nos AFDs da Figura 9 e destaca os novos recursos na interface.

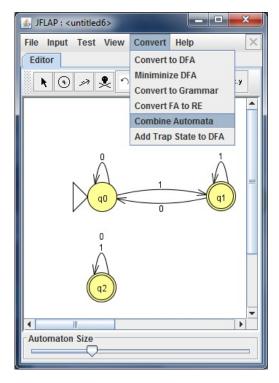


Figura 9: Seleção de dois autômatos para combinação no JFLAP [13].



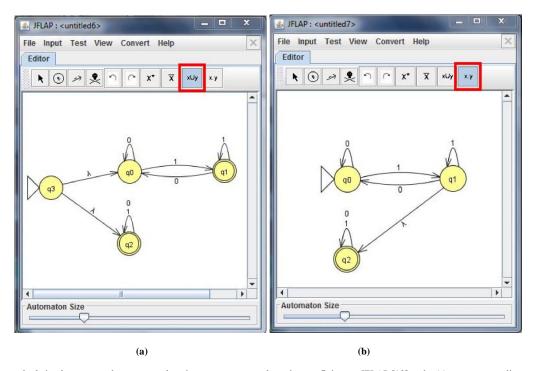


Figura 10: Exemplo da implementação de operações de união e concatenação de autômatos finitos no JFLAP [13] onde: (a) apresenta a aplicação de união e (b) apresenta a aplicação de concatenação nos autômatos da Figura 8.

Também foi solucionado outro problema de implementação do *software* original que causava confusão para usuários iniciantes e perda de tempo ao simular modelos. Ao criar uma máquina de Turing transdutora (máquina com entrada e saída) era necessário,

após realizar a computação da função, posicionar o cabeçote na primeira posição da palavra de entrada para que a saída fosse apresentada corretamente. Isso foi corrigido como apresenta a Figura 11.

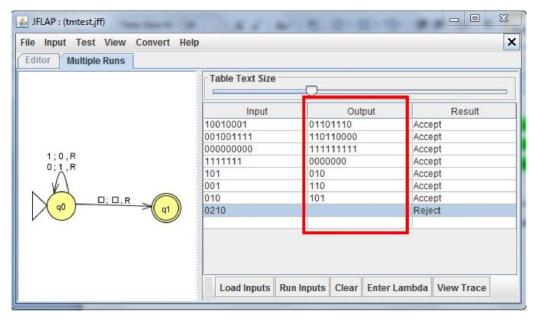


Figura 11: Exemplo de saídas de uma MT transdutora na ferramenta JFLAP [13].



5 Estudo de caso na nova versão do JFLAP

A nova versão do JFLAP, que inclui as adaptações realizadas nesse trabalho, já foi utilizada em sala de aula pela professora das disciplinas da área de LFA. A professora observou que, em geral, as adaptações possibilitaram maior agilidade no uso do *software*, além de possibilitar a realização de novas operações.

Um estudo de caso foi realizado com 17 alunos da disciplina de Linguagens Formais e Autômatos oferecida no 4º período do Bacharelado em Ciência da Computação do Centro de Ciências Agrárias da Universidade Federal do Espírito Santo – CCA/ UFES no semestre 2013/1. O objetivo do estudo de caso foi validar o trabalho desenvolvido, ou seja, verificar se as alterações incluídas no JFLAP realmente auxiliam no ensino e aprendizagem.

Os alunos participantes do estudo nunca haviam utilizado o JFLAP. Foi solicitado a eles que realizassem um conjunto de atividades propostas, (apresentadas no Apêndice B) inicialmente na versão antiga do *software* e, posteriormente, na versão adaptada (nova). Em seguida, foram apresentadas as adaptações realizadas no *software* e solicitado que eles preenchessem um questionário (apresentado no Apêndice C) sobre a experiência.

Analisando os questionários respondidos pelos alunos no estudo de caso, foi observado que 18% dos participantes observaram todas as alterações incluídas e 82% dos participantes observaram pelo menos duas alterações (Figura 12).

consideram que as alterações realizadas no JFLAP são relevantes para auxiliar o ensino e aprendizagem de LFA. No geral, pode-se resumir como as alterações realizadas auxiliaram os alunos na execução das atividades propostas, de acordo com a análise das respostas à questão 4 do questionário (Apêndice C):

- A inserção de legenda na simulação de autômatos propicia um retorno imediato para os alunos. Ela possibilita que eles saibam, a qualquer momento, o estado da computação da palavra no autômato. Isso facilita o ensino e aprendizagem da dinâmica de funcionamento do autômato, principalmente de AFN, quando várias computações são realizadas em paralelo e é possível acompanhar o estado de cada uma delas.
- A possibilidade de aplicação direta das operações estrela e complemento sobre um autômato e das operações de união e concatenação sobre dois autômatos, permite maior agilidade na construção desses modelos e na verificação da solução de



Figura 12: Gráfico com a porcentagem de alunos que observou as alterações incluídas na nova versão do JFLAP.

64% dos participantes observaram a inclusão de legenda no painel de simulação de autômatos, 88% observaram a inserção dos botões para realizar as operações estrela e complemento sobre um autômato, 52% dos participantes observaram a inserção dos botões para realizar as operações de união e concatenação sobre dois autômatos, e 94% observaram a apresentação da saída na simulação de MT transdutora. Acredita-se que um número menor de alunos tenham observado os botões de união e concatenação porque eles são apresentados apenas quando existem dois autômatos abertos, e eles tentaram fazer a partir da criação dos dois na mesma janela.

Segundo os participantes do estudo, as alterações apresentadas na nova versão auxiliaram na execução das tarefas propostas: 82% dos participantes responderam que sim e 18% respondeu que na maioria das vezes as alterações auxiliaram nas tarefas.

Todos os participantes do estudo de caso (100%)

- exercícios que envolvem essas operações. Também facilitam a demonstração dessas operações pelo professor em sala de aula.
- A apresentação da saída na MT evita a confusão gerada ao utilizar a versão antiga do software, principalmente em usuários iniciantes que esperavam ver o retorno da máquina. Ela também permite que o retorno seja visualizado rapidamente, sem ter que posicionar o cursor no primeiro símbolo da palavra, facilitando o entendimento do funcionamento da máquina.

Dos 17 alunos, 12 (cerca de 70%) afirmaram que as alterações agilizaram a verificação de exercícios. 53% afirmaram que a visualização da simulação melhorou, tanto em AFD quanto em MT.

Algumas das sugestões dadas pelos alunos foram:

- Adicionar teclas de atalho como CTRL + Z para desfazer alterações.
- Utilização da tecla ESC para cancelar uma



seleção.

• Tradução do software para o português.

6 Conclusões

Algumas correções e inclusão de funcionalidades foram realizadas na ferramenta JFLAP [13, 14, 15] com o objetivo de auxiliar o processo de ensino e aprendizagem de Linguagens Formais e Autômatos. As adaptações e inclusões de funcionalidades foram realizadas a partir de necessidades reais identificadas durante a utilização da ferramenta em sala de aula. Foram incluídas funcionalidades cuja necessidade foi detectada pela professora das disciplinas da área de LFA e funcionalidades cuja necessidade foi detectada durante o estudo de caso com alunos dessas disciplinas. Assim, foi criada uma ferramenta mais completa com o objetivo de torná-la mais usável em disciplinas da área, possibilitando o ensino mais prático e tornando a disciplina mais atrativa para os alunos.

É importante ressaltar que durante o primeiro estudo de caso os alunos reconheceram a importância da utilização do JFLAP para seu aprendizado, confirmando a importância da utilização de *softwares* como auxílio no processo de ensino-aprendizagem. E, no segundo estudo de caso, eles afirmaram que as alterações realizadas no JFLAP são relevantes para auxiliar o ensino e aprendizagem de LFA. Segundo eles, as alterações auxiliam o ensino e aprendizagem dos modelos computacionais alterados (AFD e MT) e agilizam a construção e verificação dos modelos, como apresentado na seção 5.

Futuramente, outras modificações e/ou correções podem ser realizadas a partir das sugestões dadas pelos alunos participantes dos estudos de caso. Uma delas, considerada importante, é a tradução do *software*. Além disso, outros estudos de caso podem ser realizados: com um número maior de alunos e também para avaliar a utilização das novas funcionalidades e correções adicionadas ao JFLAP.

Acredita-se que essa pesquisa possa contribuir para melhoria do ensino da disciplina Linguagens Formais e Autômatos em cursos superiores de Computação e ampliar a utilização da ferramenta JFLAP no processo de ensino e aprendizagem.

Referências

- [1] N. J. Vieira. Introdução aos Fundamentos da Computação: Linguagens e Máquinas. São Paulo: Pioneira Thomson Learning, 2006.
- [2] M. V. Ramos. Ensino de Linguagens Formais e

- Autômatos em cursos superiores de computação. ReCeT Revista de Computação e Tecnologia da PUC-SP, São Paulo, v. 1, n.1, out. 2009. http://revistas.pucsp.br/index.php/ReCET, Jul. 2013.
- [3] M. Sipser. Introdução à Teoria da Computação. 2 ed. São Paulo: Thomson Learning, 2007.
- [4] G. P. Oliveira. Construção coletiva do conhecimento através de uma experiência de incentivo à autonomia dos estudantes no aprendizado de matemática discreta. In Anais do Encontro Paulista de Educação Matemática, São Paulo, 2004.
- [5] A. S. Jukemura, A. D. Hugo, J. Q. Uchoa. GAM – Um Simulador para Auxiliar o Ensino de Teoria da Computação. In Anais do XXV Congresso da Sociedade Brasileira de Computação, São Leopoldo, páginas 2432-2443, 2005.
- [6] J. Bovet. Visual Automata Simulator: A toot for simulating, visualizing and transforming finite state automata and Turing Machines. http://www.cs.usfca.edu/~jbovet/vas.html. Jul. 2013.
- [7] C. Szymanski. Auger Ambiente para construção e simulação de autômatos finitos. http://www.evoluma.com/auger/, Jul. 2013.
- [8] SZYMANSKI, Charbel; GARCINDO, Luiz Alfredo Soares. Auger Ferramenta de apoio ao ensino de autômatos finitos. In Anais do I Workshop de Computação da Região Sul, Florianópolis SC, 2004.
- [9] N. J. Vieira. Language Emulator. http://homepages.dcc.ufmg.br/~lfvieira/ftc.html. Jul. 2013.
- [10] Vieira, L. F. M., Vieira, M. A. M., and Vieira, N. J. (2003). Language emulator, uma ferramenta de auxílio no ensino de teoria da computação. In XIII Workshop sobre Educação em Computação XXV Congresso da Sociedade Brasileira de Computação.
- [11] Y. M. G. Costa, R. C. Meneses, F. R. Uber. Software para Criação e Teste de Modelos Formais. http://www.din.uem.br/yandre/sctmf/. Jul. 2013.
- [12] Costa, Y. M. G., Meneses, R. C. e Uber, F. R.(2008). "Uma ferramenta para auxílio didático no Ensino de Teoria da Computação", em Workshop sobre Educação em Computação XXVIII Congresso da Sociedade Brasileira de Computação.
- [13] S. H. Rodger. JFLAP Version 7.0: RELEASEAD August 28, 2009.



Revista Brasileira de Informática na Educação, Volume 21, Número 3, 2013

- http://www.jflap.org, Jul. 2013.
- [14] S. Rodger, T. Finley. JFLAP An Interactive Formal Languages and Automata Package. ISBN 0763738344, Jones and Bartlett, 2006.
- [15] P. Chakraborty, P.C. Saxena, and C. P. Katti, Fifty years of automata simulation: a review, ACM Inroads, Volume 2, Issue 4, p. 59-70, December 2011.
- [16] R. S. Aguiar, J. Y. Y. Oeiras. Laboratório de Linguagens Formais LabLF. Revista Brasileira de Informática na Educação. 18(1): 108-111, 2010.



APENDICE A

Questionário preenchido pelos participantes do estudo de caso apresentado na seção 3

Caro usuário, este questionário tem como objetivo extrair informações para o projeto de pesquisa "Estudo e adaptação de *software* para o ensino de Linguagens Formais e Autômatos". Por favor, responda as questões a seguir sendo o mais sincero possível. Suas sugestões na questão 8 serão de grande importância para o nosso estudo. Agradecemos a sua colaboração.

1)	Há quanto tempo você utiliza o software JFLAP?
	() Há mais de dois anos.
	() De 1 a 2 anos.
	() De 6 meses a 1 ano.
	() Menos de 6 meses.
2)	Que tipo de usuário do JFLAP você se considera?
	() Usuário avançado.
	() Usuário intermediário.
	() Usuário iniciante.
3)	Qual das opções abaixo melhor descreve sua 1ª experiência utilizando o JFLAP?
	() Consegui usar o JFLAP sem dificuldades.
	() Consegui usar o JFLAP após ajuda de um colega ou do professor.
	() Não consegui utilizar o JFLAP.
	() Não me lembro.
4)	A documentação oferecida pelo JFLAP foi suficiente para sanar as dúvidas que surgiram durante o seu uso?
	() Sim.
	() Na maioria das vezes.
	() Poucas vezes.
	() Não.
	() Nunca recorri a documentação.
5)	Qual a importância do JFLAP no seu aprendizado da(s) disciplina(s) em que foi utilizado?
	() Muito importante.
	() Importante.
	() Pouco importante.
	() Nada importante.
6)	O fato da interface do JFLAP ser em língua estrangeira, prejudicou uma melhor compreensão do programa?
	() Sim.
	() Na maioria das vezes.
	() Poucas vezes.
	() Não.
7) dúvida	Alguma vez durante a utilização do JFLAP, ele apresentou algum comportamento não esperado ou que gerasse a?
	() Sim.



Revista Brasileira de Informática na Educação, Volume 21, Número 3, 2013

) S1	ua resposta seja sim, descreva o que ocorreu.
_	
_	
_	
_	
Sug	gestões (adaptações ou funcionalidades que podem ser inseridas no JFLAP):
_	



APENDICE B

Atividades propostas aos participantes do estudo de caso apresentado na seção 5

Utilizando a ferramenta JFLAP, faça o que se pede:

- 1) Construa um AFN para a linguagem $L = \{x01y \mid x, y \in \{0,1\}^*\}$. Em seguida, utilize a opção Input \rightarrow Step with Closure para executar a simulação do comportamento do autômato para algumas palavras de entrada. Observe se ele está funcionando corretamente.
- 2) Construa agora um AF para o complemento da linguagem do exercício 1. Lembre-se que para fazer o complemento deve-se utilizar um AFD. Utilize a opção Input → Multiple Run para executar a simulação do comportamento do autômato para várias palavras de entrada ao mesmo tempo. Observe se ele está funcionando corretamente.
- 3) Construa um AFN para a união das linguagens $L1 = \{x01y \mid x, y \in \{0,1\}^*\}$ e $L2 = \{w \mid w \in \{0,1\}^*\}$. Você pode abrir um AF salvo e utilizar a opção Convert \rightarrow Combine Automata para combinar dois autômatos construídos previamente em janelas diferentes.
- 4) Construa uma MT transdutora que receba como entrada palavras sobre $\Sigma = \{0,1\}$ e troque todos os 0s por 1s e vice-versa. Utilize a opção Input \rightarrow Multiple Run para executar a simulação da MT. Verifique se ela está funcionando corretamente.



APENDICE C

Questionário preenchido pelos participantes do estudo de caso apresentado na seção 5

Caro usuário, este questionário tem como objetivo extrair informações para o projeto de pesquisa "Estudo e adaptação de software para o ensino de Linguagens Formais e Autômatos". Por favor, responda as questões a seguir sendo o mais sincero possível. Suas sugestões na questão 5 serão de grande importância para o nosso estudo. Agradecemos a sua colaboração.

1)	Ao realizar as atividades propostas, você observou as alterações inseridas na nova versão do JFLAP? () Sim, observei todas as alterações.						
	() Não observei nenhuma alteração.						
	() Sim, observei algumas alterações.						
	Se marcou a última opção (observou algumas alterações), indique quais alterações foram observadas:						
	() Inserção de legenda no painel de simulação de autômatos finitos.						
	() Inserção de dois novos botões na barra de ferramentas do editor de autômatos finitos: X^* e X que executam as operações estrela e complemento de um autômato finitorespectivamente.						
	() Inserção de dois novos botões na barra de ferramentas do editor de autômatos finitos ao combinar dois autômatos: X U Y e X.Y que executam as operações de união e concatenação de dois autômatos finitos respectivamente.						
	() Apresentação da saída (output) na simulação de MT transdutora usando Multiple Run.						
2)	As alterações apresentadas na nova versão do JFLAP auxiliaram na execução das tarefas propostas? () Sim. () Na maioria das vezes. () Poucas vezes. () Não, porque não foram observadas. Se tivessem sido observadas, teriam ajudado. () Não.						
3)	Você considera as alterações realizadas no JFLAP relevantes para auxiliar no ensino/aprendizagem da disciplina Linguagens Formais e Autômatos. () Sim.						
	() Não.						
4)	Explique como as alterações realizadas auxiliaram a execução das tarefas propostas:						
5)	Observações e/ou sugestões:						