UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Campus Presidente Prudente

Linguagens Formais e Autômatos

Automatos Finitos

Versão 1.0.2

Pétrus Antonio Barbosa Pradella
Luiz Henrique Takayuki de Utimura Zorzatto

Presidente Prudente
Novembro de 2018

Sumário

1 Propósito deste Documento

Esse documento tem como intuito explicar como funciona o programa FiniteAutomation.

FiniteAutomation é um programa feito em Java, criado durante a disciplina de "Autômatos Finitos"como projeto semestral.

FiniteAutomation é um compilado de diversos programas juntos em um único software, programas esses cada qual com suas respectivas funções e finalidades.

2 RegexTool

2.1 O Programa

O programa usa da tecnologia JavaFX para gerar as interfaces gráficas (Scene Builder).

2.2 Instruções

O usuário deve inserir no primeiro campo a regex desejada, e no segundo campo a expressão que ele quer validar, e por fim clicar no botão validar!

Para converter um AutomatoFinito para Regex deve-se clicar no botão indicado "Converter De AF para Regex"

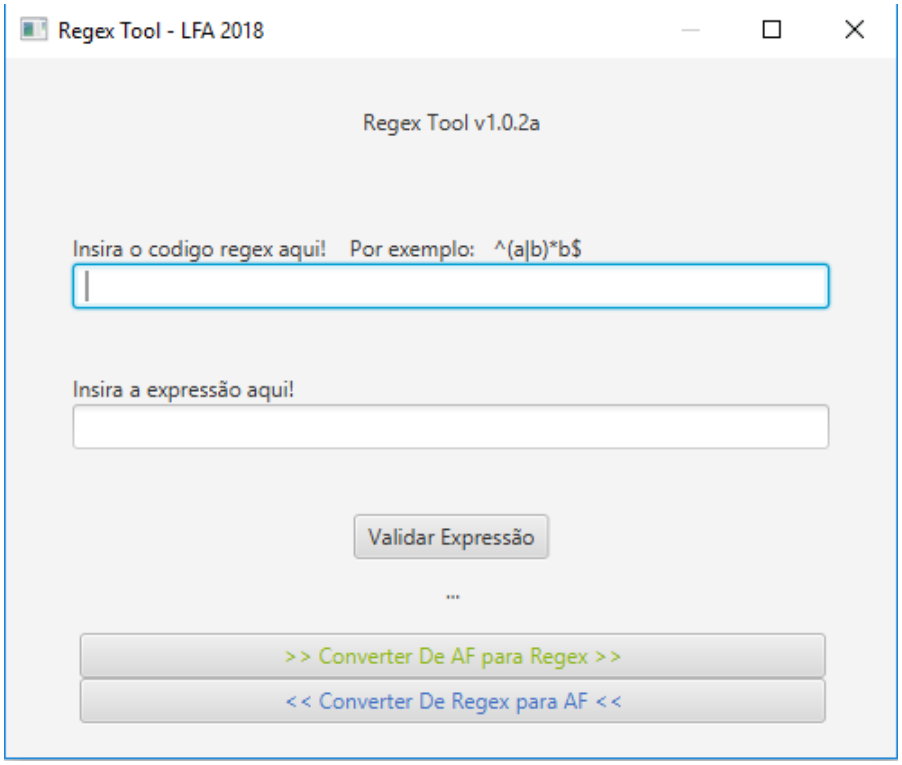

Figura 1: RegexTool

3 GrammarTool

3.1 O Programa

3.1.1 Interface

O programa usa da tecnologia JavaFX para gerar as interfaces gráficas (Scene Builder).

3.1.2 Codigo

O código desse programa utiliza de recursividade para resolver a gramatica desejada. O programa se constitui basicamente de 3 classes:

1 - GrammarExpression.class - guarda o conteúdo das gramaticas.

2 - GrammarValidator.class - valida uma expressão de acordo com a gramatica inserida.

3 - ValidadeData.class - auxilia na recursão durante a validação.

3.2 Instruções

3.2.1 Inserindo e Removendo Gramaticas

O usuário deve inserir no primeiro campo de texto a letra que corresponde ao símbolo não terminal, e a direita desse mesmo campo de texto deve inserir qual é o conjunto de símbolos que esse símbolo não terminal representa (em seguida, clicar no botão "Adicionar Expressão).

Para remover uma expressão inserida, basta clicar naquela que deseja remover e clicar no botão "Apagar Expressão Selecionada".

3.2.2 Validando uma Expressão

Para validar uma expressão de acordo com as regras gramaticais inseridas, basta escreve a expressão no terceiro campo de texto e clicar em "Validar Texto Segundo Gramática".

3.2.3 Conversão

Para converter entre Automato Finito e Gramática é necessário clicar nos botões no final da tela.

- "Converter De AF para Gramatica"
- "Converter De Gramatica para AF"

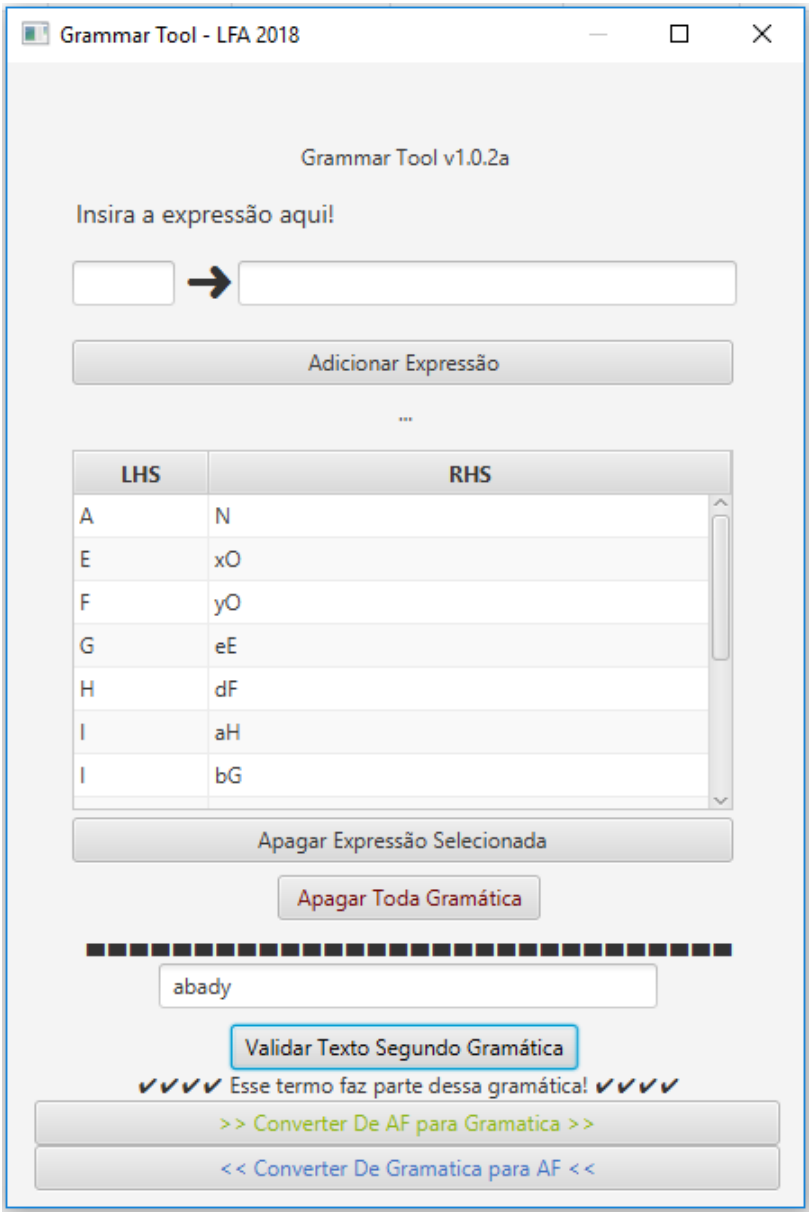

Figura 2: Grammar Tool

4 Automato Finito

4.1 O Programa

4.1.1 Interface

O programa usa da tecnologia JavaFX para gerar as interfaces gráficas (Scene Builder). Além disso, parte do código do professor Danilo é utilizado para realizar a criação de círculos e arestas.

4.1.2 Código

O código desse programa utiliza de recursividade para resolver o autômato finito. A resolução do autômato em si se compõe apenas de 4 classes.

1 - Vertice.class - determina um estado do autômato finito.

2 - Aresta.class - guarda o conteúdo das transições entre vértices.

3 - Validator.class - valida uma expressão de acordo com o autômato finito.

4 - HistoryLog.class - auxilia na recursão durante a validação. (Além de salvar um histórico dos movimentos realizados);

As funções de "Remoção De Arestas Nulas"e "Conversão de Automato Finito Nâo Deterministico para Automato Finito Deterministico"estão contidas na classe Validator.class.

4.2 Instruções

4.2.1 Importando/Exportando arquivos

Para importar ou exportar algum automato finito, clique na aba "Arquivo"e selecione sua respectiva função.

Especifique o nome do arquivo de destino ou de origem (de acordo com qual das funções você deseja realizar).

O arquivo de origem para um carregamento deve estar necessáriamente na mesma pasta que o programa está sendo executado.

A pasta de destino dos arquivos salvos necessáriamente será a pasta em que o programa está seundo executado!

Você pode usar o mouse para clicar e arrastar os estados.

Para editar um estado (deletar, mudar o nome, marcar como final ou inicial) basta clicar com o botão direito no mesmo.

4.2.2 Inserindo, movendo, removendo e alterando Estados

Com a RadioBox "Mover/Editar Estados"selecionada.

Para adicionar estados basta clicar no botão "Adicionar Estado", assim um novo estado será adicionado ao centro da janela.

Você pode usar o mouse para clicar e arrastar os estados.

Para editar um estado (deletar, mudar o nome, marcar como final ou inicial) basta clicar com o botão direito no mesmo.

4.2.3 Criando e alterado Transições

Com a RadioBox "Editar Transições"selecionada.

Para adicionar uma transição basta clicar em um estado, e mantendo o botão do mouse pressionado, arrastar a linha até um outro estado de destino.

Para editar uma transição (deletar conteúdo) basta clicar com o botão direito na mesmo.

Nota: O program irá anular redundancia automaticamente, ou seja, não irá aceitar que uma regra seja adicionada duas vezes a mesma transição.

Alem disso, você pode adicionar várias regras ao mesmo tempo. Por exemplo, se você adicionar a regra "abcdefg"ele irá adicionar "a$\|b|c|d|e|f|g$".

4.2.4 Validando Expressões

Para validar expressões, basta clicar no botão "Testar Autômato".

Na janela aberta, existe três possibilidades para testes.

O primeiro campo é o de teste direto, basta inserir um texto e clicar no botão "Teste Direto".

O segundo campo é o de teste múltiplo, basta inserir diversas expressões, sendo uma expressão por linha, e clicar no botão "Teste Múltiplo".

O terceiro campo é o de teste passo a passo, basta inserir a expressão, e se ela for válida, os botões de ir para frente e para trás ficarão liberados.

(Ao clicar neles, é possível observar o movimento simulado que é realizado no autômato).

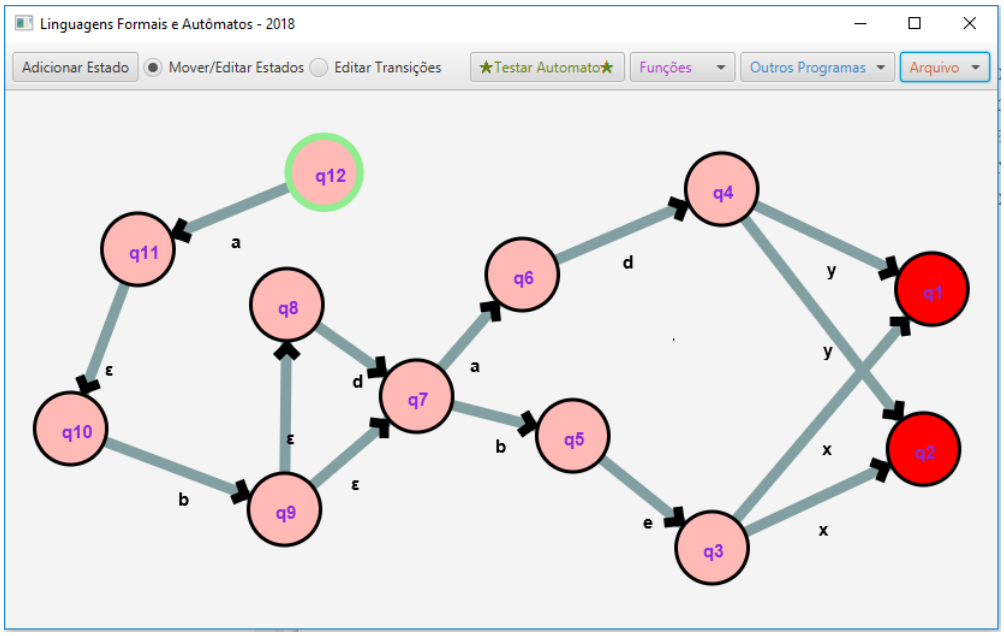

Figura 3: Autômato Finito (página inicial)

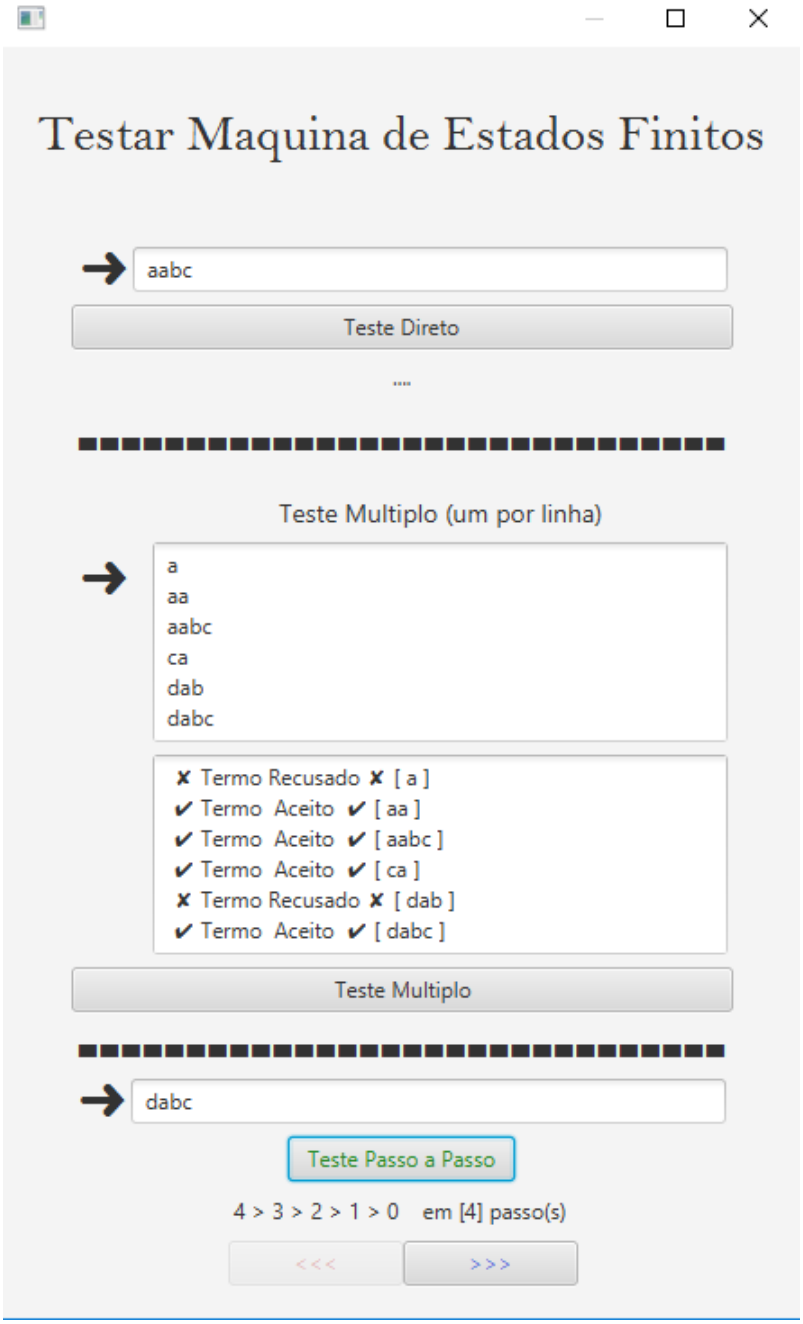

Figura 4: Autômato Finito (validar expressão)

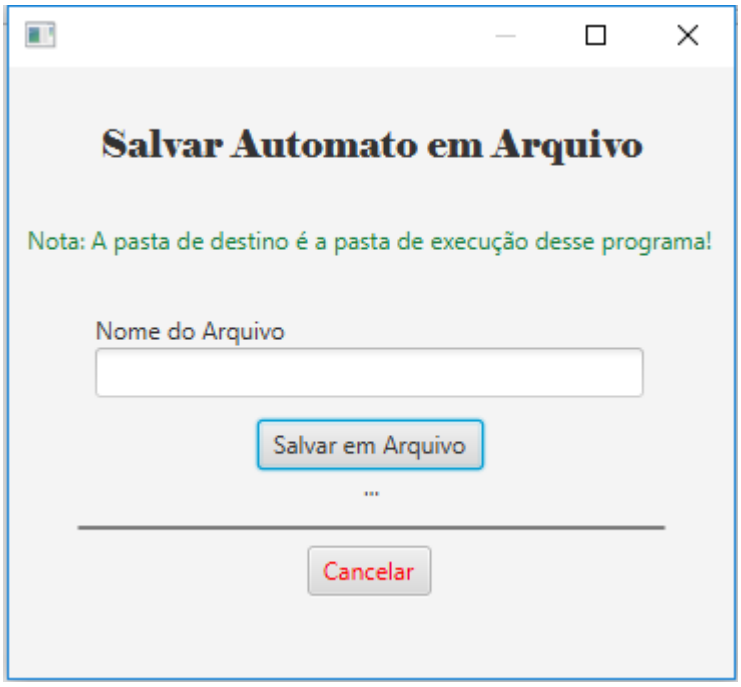

Figura 5: Autômato Finito (validar expressão)