

**UNIVERSIDADE ESTADUAL DE LONDRINA**  
**LINGUAGENS FORMAIS E AUTÔMATOS**

**Rafael Palheta Tokairin**

**ANÁLISE DO SOFTWARE JFLAP**  
**UM ESTUDO SOBRE SUAS APLICAÇÕES E LIMITAÇÕES**

**Trabalho apresentado na matéria  
de Linguagens Formais e  
Autômatos**

**Orientadora: Cinthyan Renata  
Sachs Camerlengo de Barbosa**

**Londrina  
2024**

# Software JFLAP

## Introdução

O JFLAP é um software educacional interativo escrito em Java para experimentação em tópicos na área de ciência da computação de linguagens formais e teoria dos autômatos, principalmente destinado ao uso no nível de graduação ou como um tópico avançado para o ensino médio.

A imagem abaixo ilustra a interface do programa:

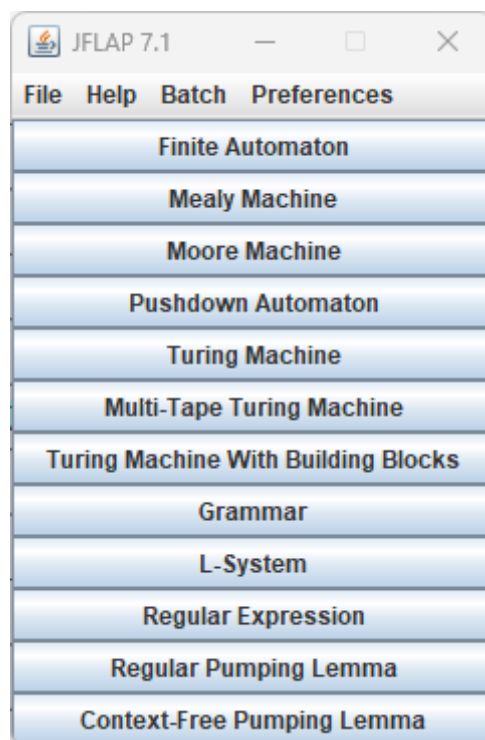


Figura 1: JFLAP 7.1 - Menu. Fonte: O autor.

De acordo com as opções de interface, o software permite a representação de:

- Autômatos Finitos e de Pilha;
- Máquinas de Mealy, Moore, Turing, Turing com múltiplas cabeças e Turing com caixas-pretas;
- Gramáticas;
- Expressões Regulares;
- Lema do Bombeamento de Gramáticas Livres de Contexto e Regulares.

Para essa avaliação serão utilizadas as ferramentas para Gramáticas, Autômatos Finitos e de Pilha, Máquinas de Mealy, Moore e Expressões Regulares.

## Interface

A interface do programa é construída utilizando a biblioteca Java padrão. No entanto, devido a certas decisões de design, a interface pode parecer antiquada em comparação com as plataformas mais recentes. O tamanho do texto muitas vezes não se ajusta bem às resoluções dos monitores modernos, resultando em texto de tamanho reduzido. Além disso, a utilização de abreviações como DFA, RE e FA pode representar um obstáculo, especialmente para estudantes iniciantes no uso do software, e ainda mais para aqueles com conhecimento limitado em inglês ou que não o têm como língua principal. Adicionalmente, algumas funções importantes podem parecer escondidas na interface, exigindo a leitura da documentação para serem encontradas.

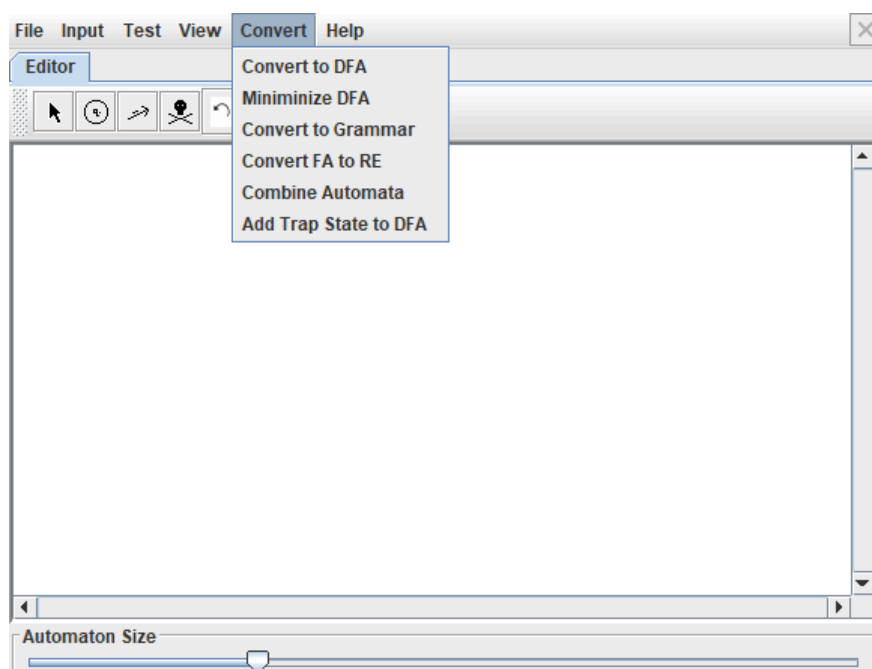


Figura 2: JFLAP 7.1 - Interface da aba de autômatos finitos. Fonte: O autor.

## Testes no JFLAP com exemplos dos Slides vistos em sala de aula

Dentro do JFLAP, há a opção de acessar a aba "Finite Automaton", onde os usuários podem inserir autômatos desejados. Após isso, é possível utilizar a função "Step by State" ou "Multiple Run" para determinar se a cadeia de símbolos é aceita. Essa ferramenta simplifica a análise e teste de cadeia de símbolos de um autômato.

Contudo, a abordagem de "Multiple Run" empregada para verificar múltiplas cadeia de símbolos, analisando cada uma, pode acarretar em lentidão e demora na obtenção dos resultados desejados.

# Slide 13

## Exemplo 2.1

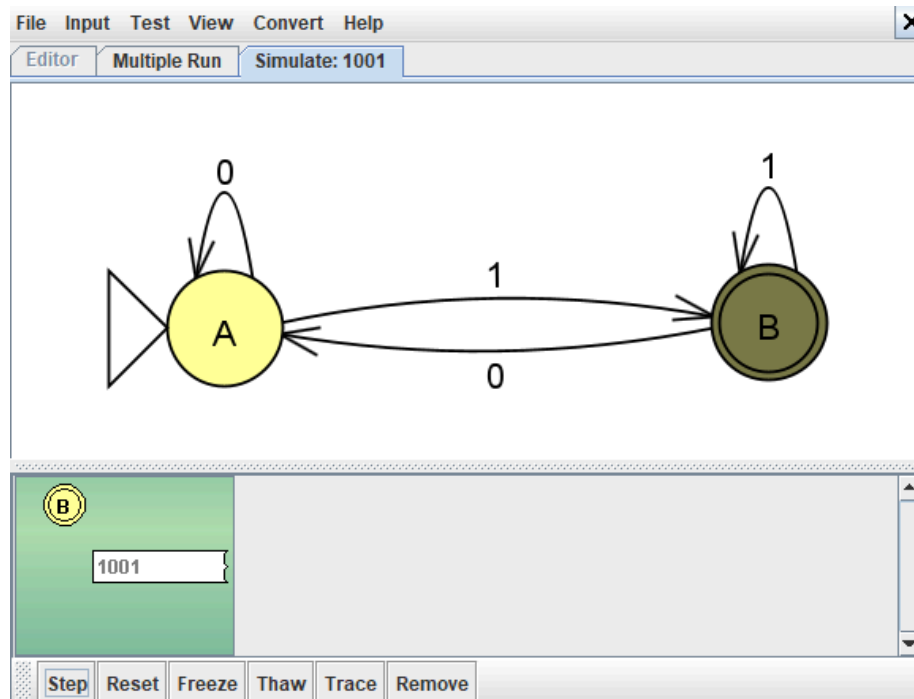


Figura 3: JFLAP 7.1 - Análise de cadeia de símbolos passo a passo. Fonte: O autor.

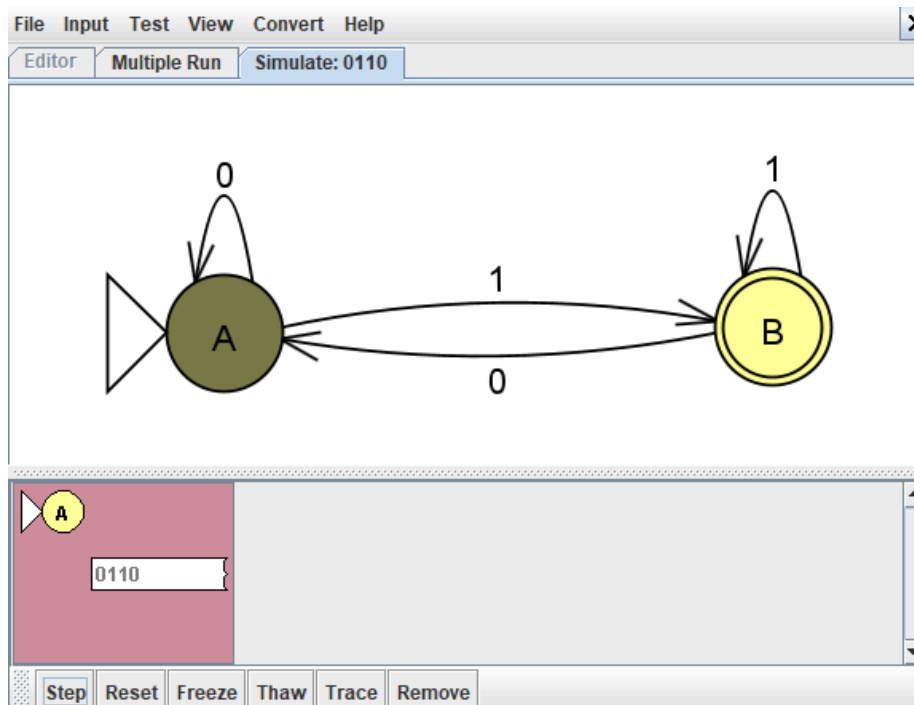


Figura 4: JFLAP 7.1 - Análise de cadeia de símbolos passo a passo. Fonte: O autor.

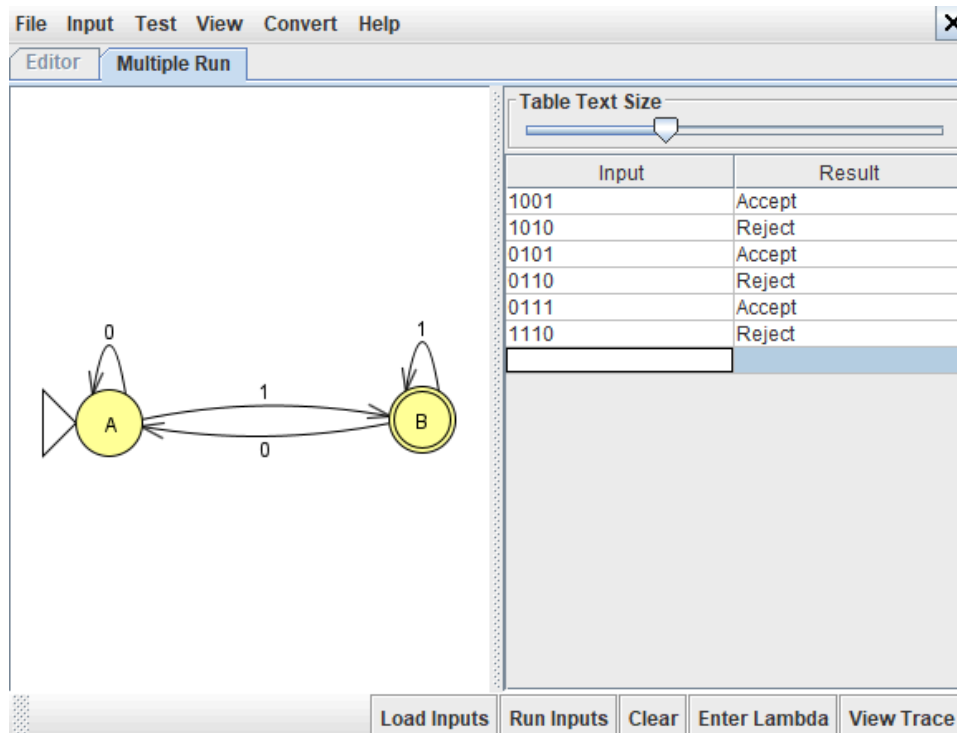


Figura 5: JFLAP 7.1 - Análise de cadeia de símbolos com execuções múltiplas.  
Fonte: O autor.

O software permite a criação de Autômatos Finitos Determinísticos e não-Determinísticos.

## Exemplo 2.2

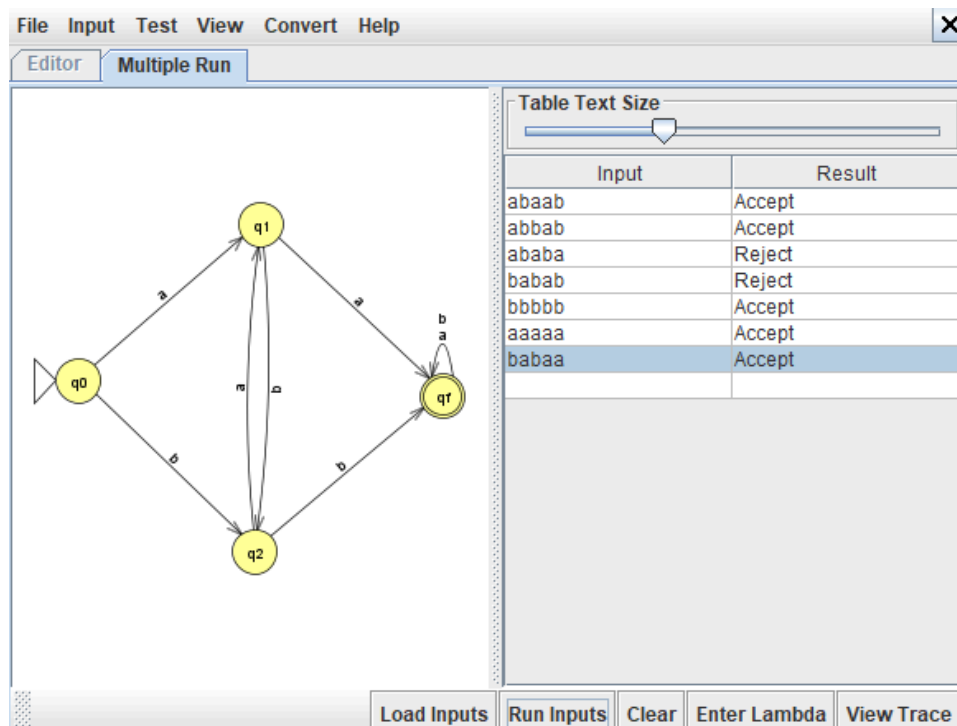


Figura 6: JFLAP 7.1 - Análise de cadeia de símbolos com execuções múltiplas em Autômato Finito Determinístico. Fonte: O autor.

### Exemplo 2.3

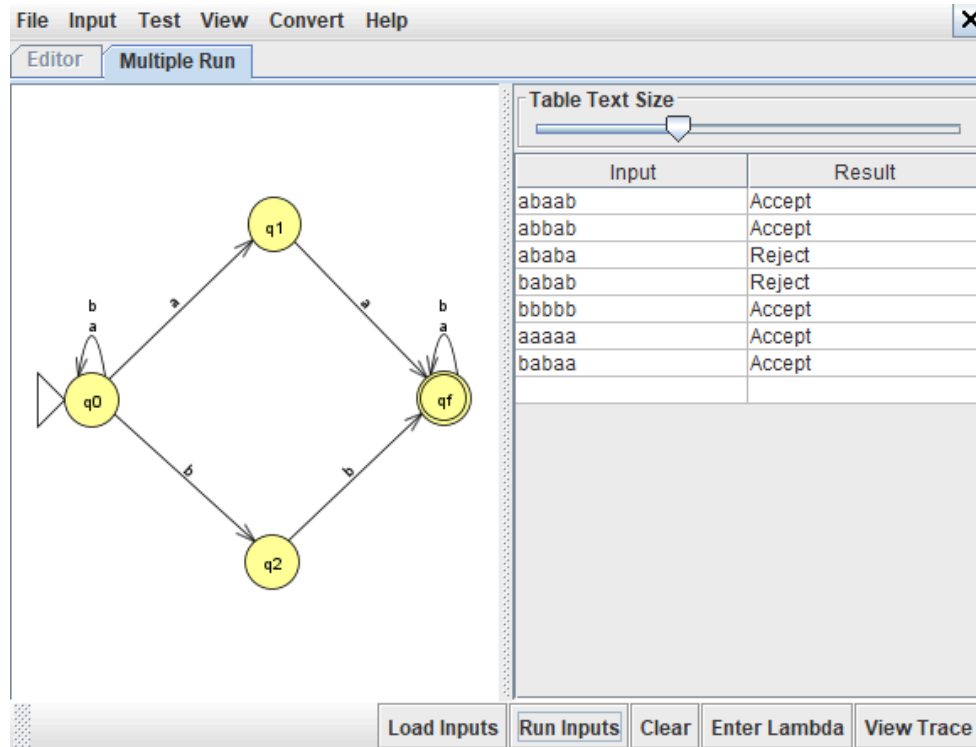


Figura 7: JFLAP 7.1 - Análise de cadeia de símbolos com execuções múltiplas em Autômato Finito não-Determinístico. Fonte: O autor.

## Slide 15

### Teorema 2.1, Exercício

O software é capaz de converter um Autômato Finito não-Determinístico para um Autômato Finito Determinístico, na aba "NFA to DFA" nas ferramentas para Autômatos Finitos, assim como o Teorema 2.1 determina.

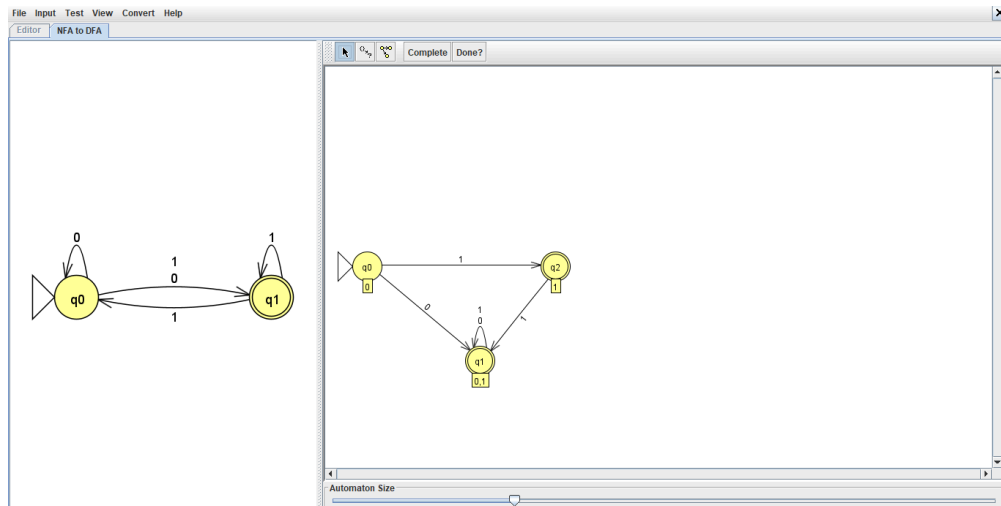


Figura 8: JFLAP 7.1 - Conversão Autômato Finito não-Determinístico para Autômato Finito Determinístico. Fonte: O autor.

## Slide 16

### Teorema 2.2, Exemplo 2.4

A partir de uma Gramática Regular, o software converte para Autômato Finito, na aba “Convert to FA” nas ferramentas para Gramáticas, assim como o Teorema 2.2 determina.

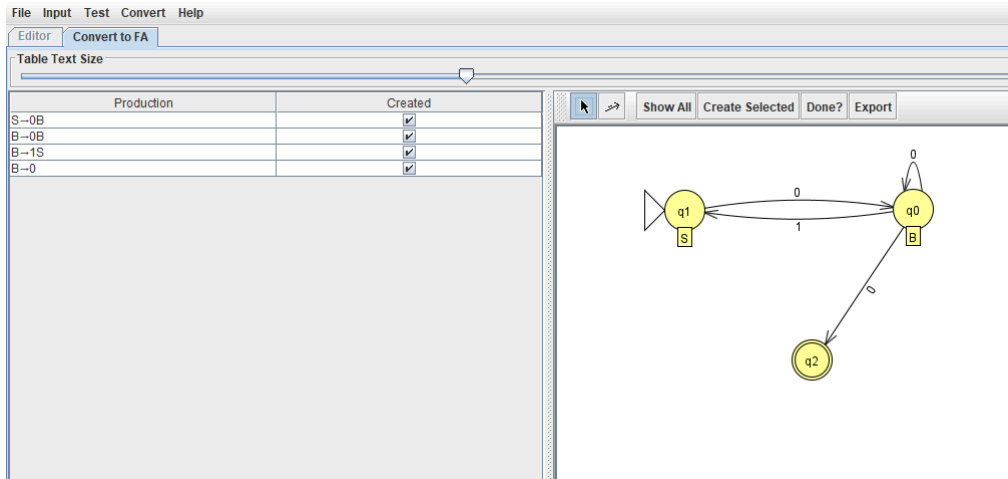


Figura 9: JFLAP 7.1 - Conversão Gramática Regular para Autômato Finito. Fonte: O autor.

### Teorema 2.3, Exemplo 2.4

A partir de um Autômato Finito, o software converte para Gramática Regular, na aba “Convert to Grammar” nas ferramentas para Autômatos Finitos, assim como o Teorema 2.3 determina.

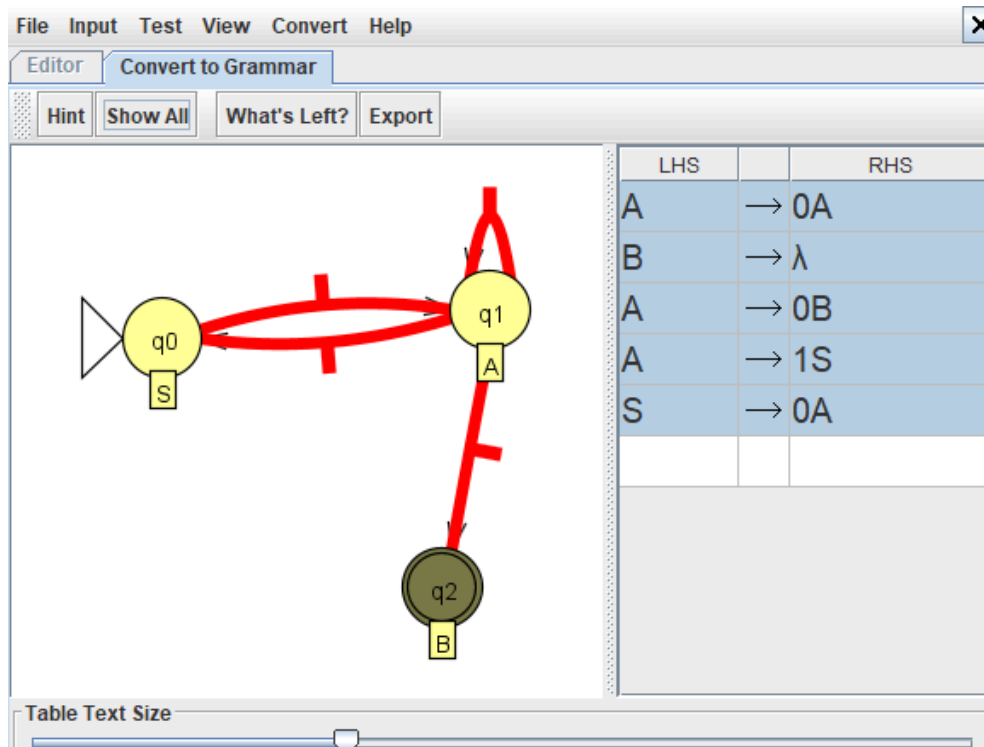


Figura 10: JFLAP 7.1 - Conversão Autômato Finito para Gramática Regular. Fonte: O autor.

### Definição 2.5, Exemplo 2.4

O software permite a conversão de Autômato Finito para Expressão Regular e Expressão Regular para Autômato Finito não-Determinístico, na aba "Convert FA to RE" e "Convert RE to NFA" nas ferramentas para Autômatos Finitos, assim como a Definição 2.5 determina.

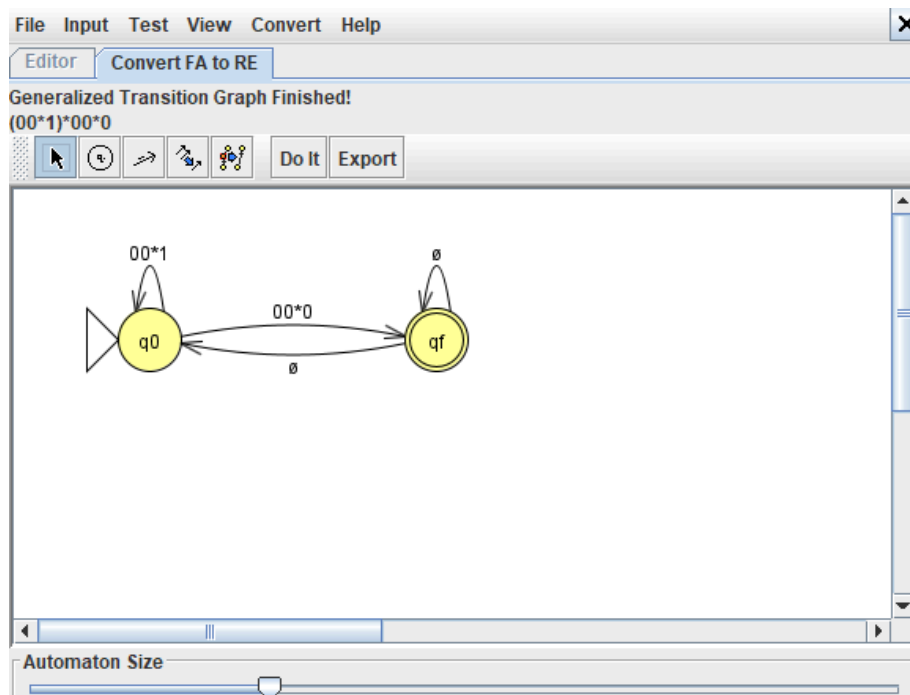




Figura 11: JFLAP 7.1 - Conversão de Autômato Finito para Expressão Regular.  
Fonte: O autor.

## Exemplo 2)

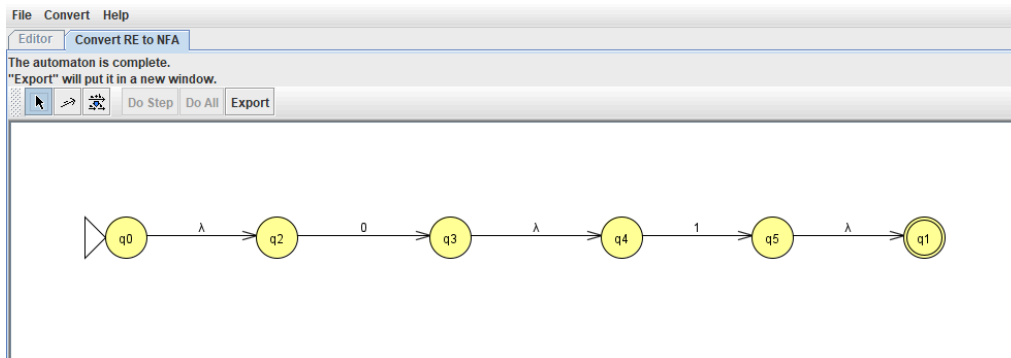


Figura 12: JFLAP 7.1 - Conversão de Expressão Regular para Autômato Finito não-Determinístico. Fonte: O autor.

## Slide 17

### Exercícios 1) e 2)

O software não mostra a Regra de Arden sendo aplicada, porém é fácil encontrar a Expressão Regular dada uma gramática, basta converter a Gramática para Autômato Finito, depois clicar na aba "Export" que irá exportar o Autômato para as ferramentas Autômatos Finitos, em seguida converta para expressão regular.

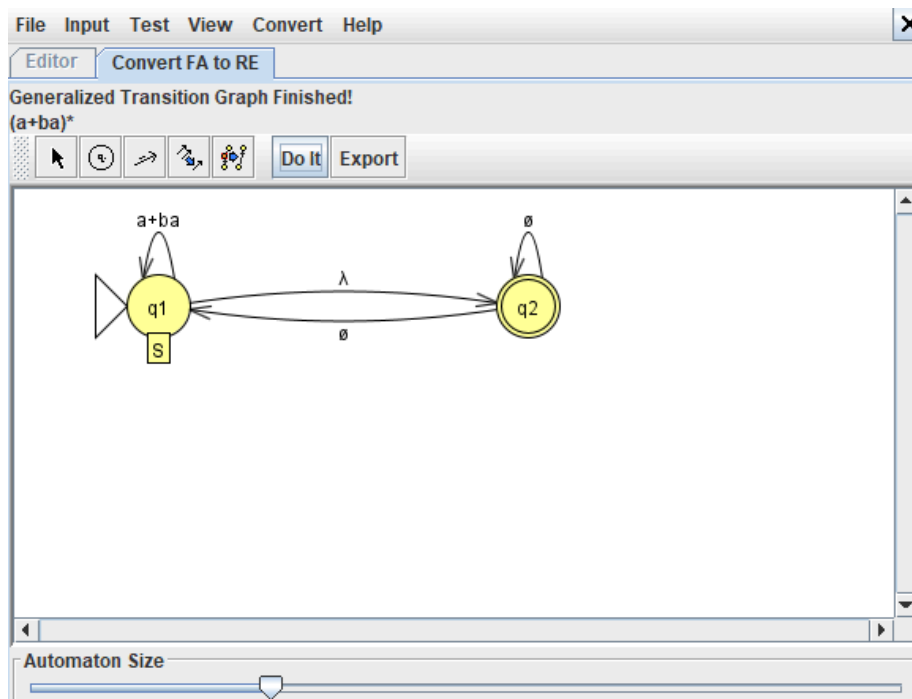


Figura 13: JFLAP 7.1 - Conversão para Expressão Regular. Fonte: O autor.

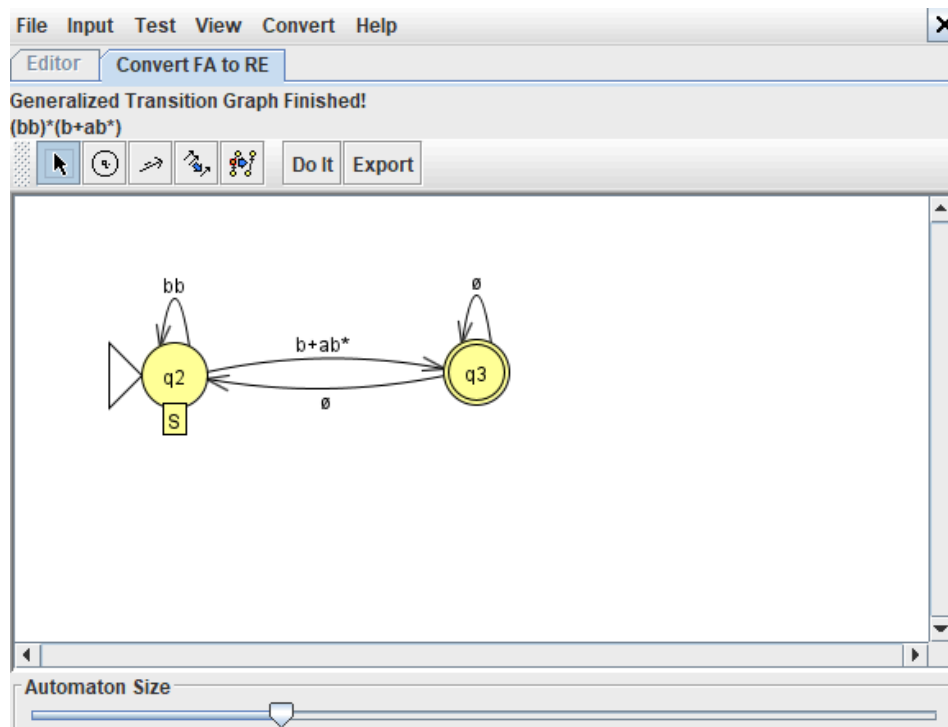


Figura 14: JFLAP 7.1 - Conversão para Expressão Regular. Fonte: O autor.

Não foram encontradas maneiras de se utilizar o Método Indutivo e o Método das Equações Simultâneas presentes no slide 18.

## Slide 19

### Exercícios passados em sala de aula

O JFLAP permite a construção da Máquina de Mealy e de Moore e gera a palavra de saída para as transições dadas, foi testado um exemplo passado em sala de aula para verificar os Teorema 2.4 e 2.5 que mostra, assim, a equivalência entre ambas.

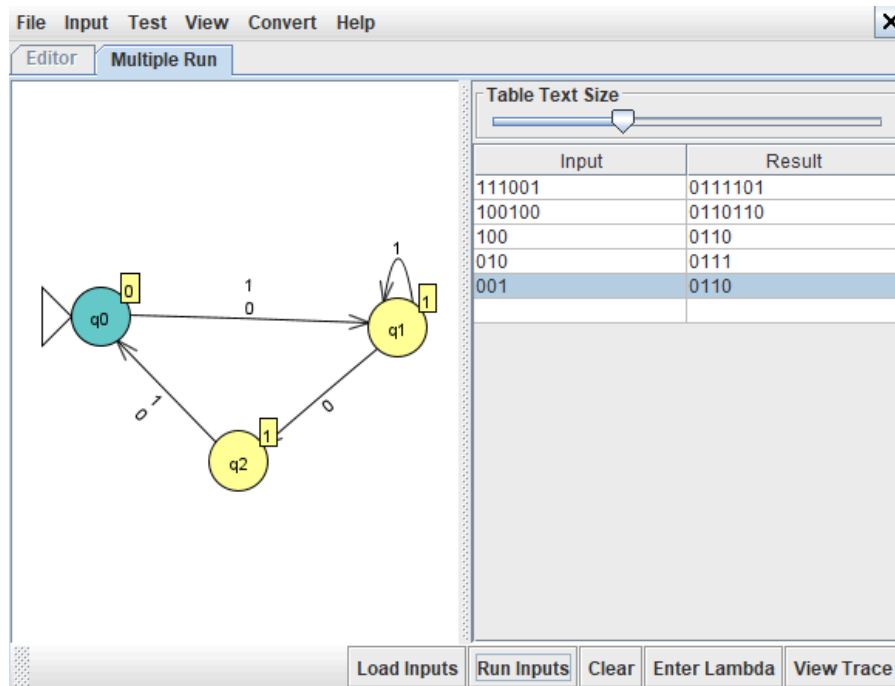


Figura 15: JFLAP 7.1 - Máquina de Moore. Fonte: O autor.

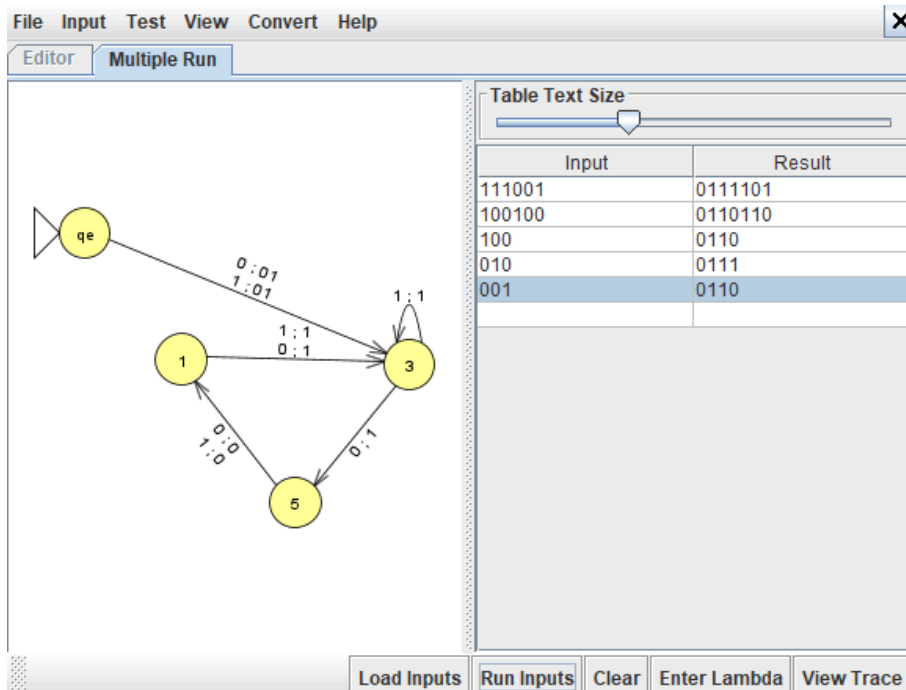


Figura 16: JFLAP 7.1 - Máquina de Mealy. Fonte: O autor.

## Slide 20

### Exemplo 2.9

O software permite a criação de Autômatos de Pilha, podemos testar etapa por etapa ou múltiplas palavras utilizando a verificação por meio de pilha vazia ou se alcançou o estado final, porém não é possível utilizar "?", para indicar o teste de

pilha vazia ou toda palavra de entrada lida, portanto não foi possível demonstrar os exemplos do Slide 21.

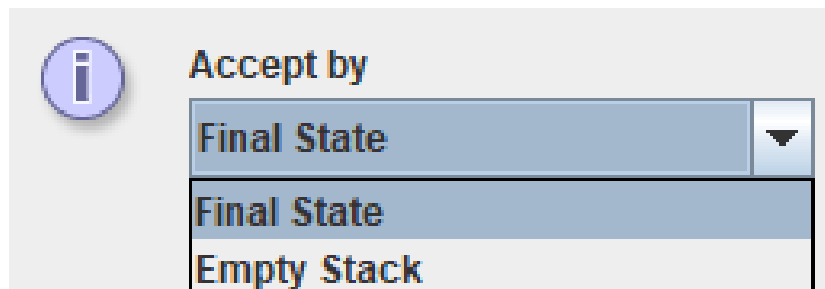


Figura 17: JFLAP 7.1 - Verificação por meio de pilha vazia ou estado final. Fonte: O autor.

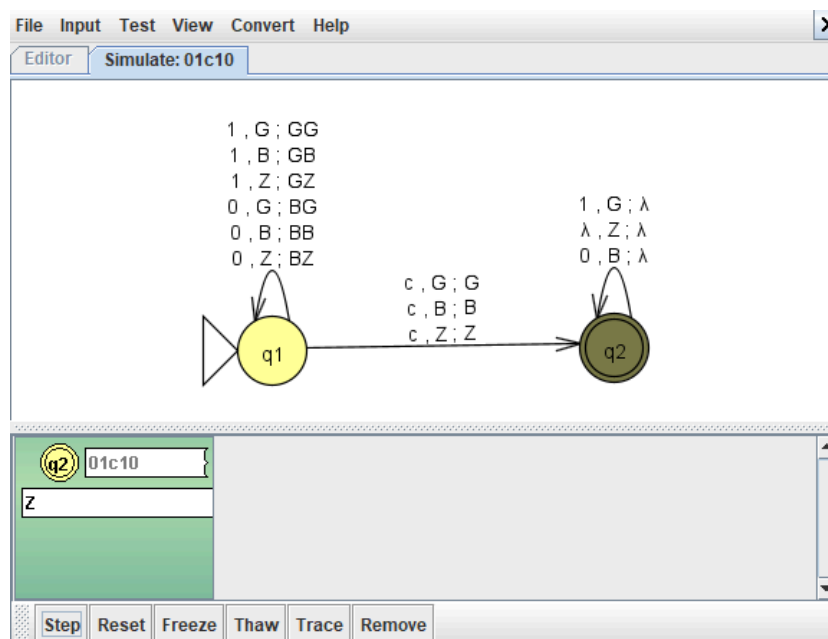


Figura 18: JFLAP 7.1 - Exemplo 2.9 utilizando etapa por etapa. Fonte: O autor.

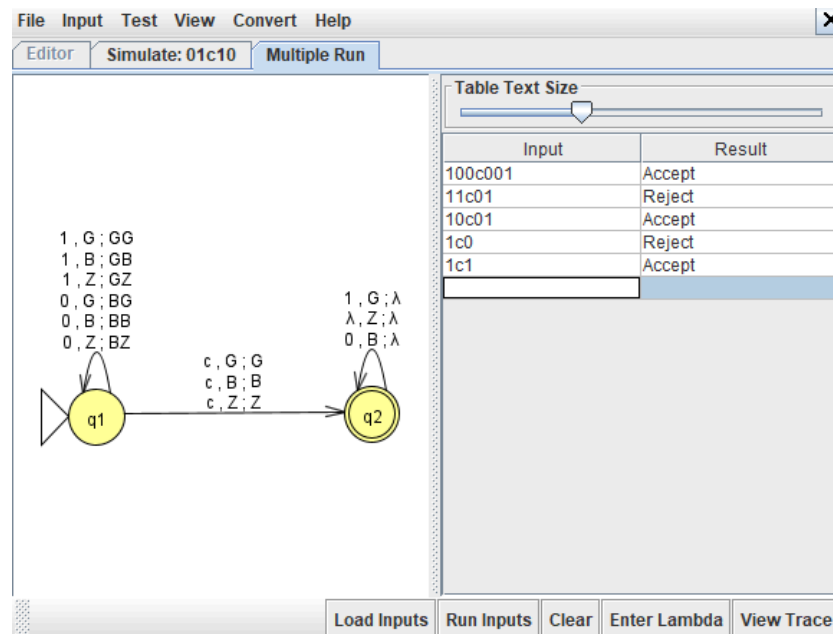


Figura 19: JFLAP 7.1 - Exemplo 2.9 utilizando múltiplas palavras. Fonte: O autor.

## Conclusão

### Pontos positivos

O JFLAP pode simular autômatos finitos para modelar sistemas de estados e transições, além de explorar autômatos de pilha para lidar com gramáticas livres de contexto. As Máquinas de Mealy e Moore são facilmente implementadas, permitindo a análise de sistemas reativos e a compreensão das saídas em função das entradas. Além disso, o JFLAP oferece suporte para a manipulação de Expressões Regulares, facilitando a busca e manipulação de padrões em textos e linguagens formais.

### Pontos negativos

Uma limitação significativa é a lentidão do método “Multiple Run” para testar a validade de palavras que podem impactar a eficiência e a experiência do usuário.

### Possíveis melhorias

Para melhorar o software, seria importante considerar uma interface mais moderna e ajustável a diferentes resoluções de tela. Além disso, o aprimoramento da velocidade de processamento, especialmente em operações que demandam

“Multiple Run”, poderiam aumentar significativamente a usabilidade e a eficácia do JFLAP.

## Resultado

Apesar de suas limitações, o JFLAP tem desempenhado um papel importante no estudo de disciplinas como Linguagens Formais e Autômatos, fornecendo ferramentas essenciais para o aprendizado e a pesquisa nessa área.