

Alunos: Thomas Fernandes Feijoo (12200662)
Fabio Volkmann Coelho (15200589)

Trabalho Prático 1 - Relatório

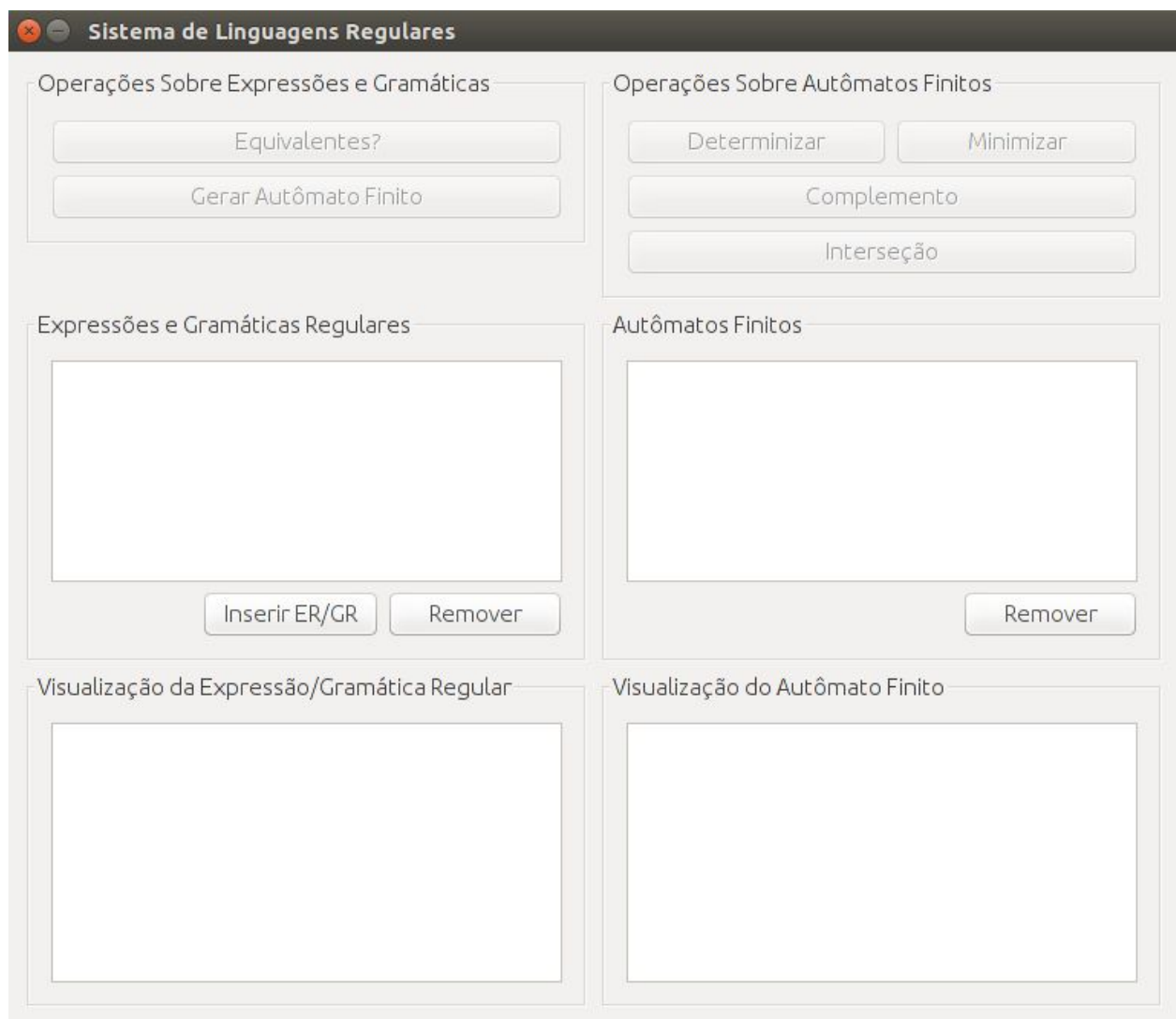
1. Tecnologias utilizadas

O trabalho foi feito utilizando a linguagem de programação Java, em sua versão 7, fazendo uso do paradigma de orientação a objetos e do padrão MVC.

2. Utilização

É necessário ter o Java 7 instalado no computador para poder utilizar o sistema. Tendo a versão correta instalada, o programa pode ser rodado via terminal através do comando “java -jar trabalho1-2018.jar”.

Ao executar o programa, esta tela será aberta:



Para inserir uma nova Expressão Regular ou uma Gramática Regular, o usuário deve clicar no botão “Inserir ER/GR”. Após o click, uma nova janela irá abrir onde ele poderá fazer a inserção desejada.

The image displays two instances of a dialog box titled "Inserção/Edição de Dispositivos Regulares".

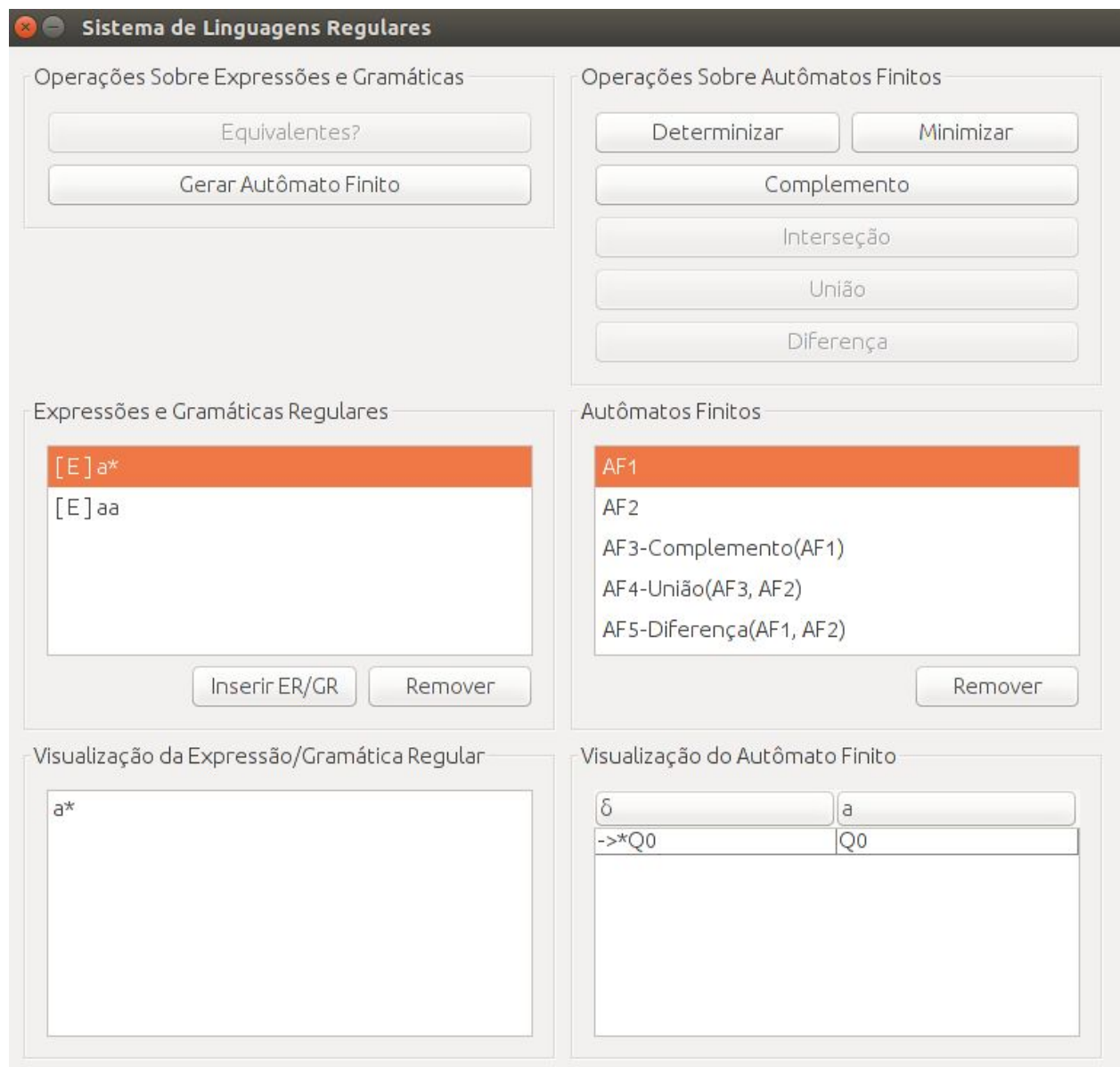
Left Dialog Box:

- Tipo do Dispositivo:** ☒ Expressão Regular ☐ Gramática Regular
- Descrição Textual:** A text area containing the regular expression `(ab)*cd|`.
- Buttons:** "Cancelar" and "Ok".

Right Dialog Box:

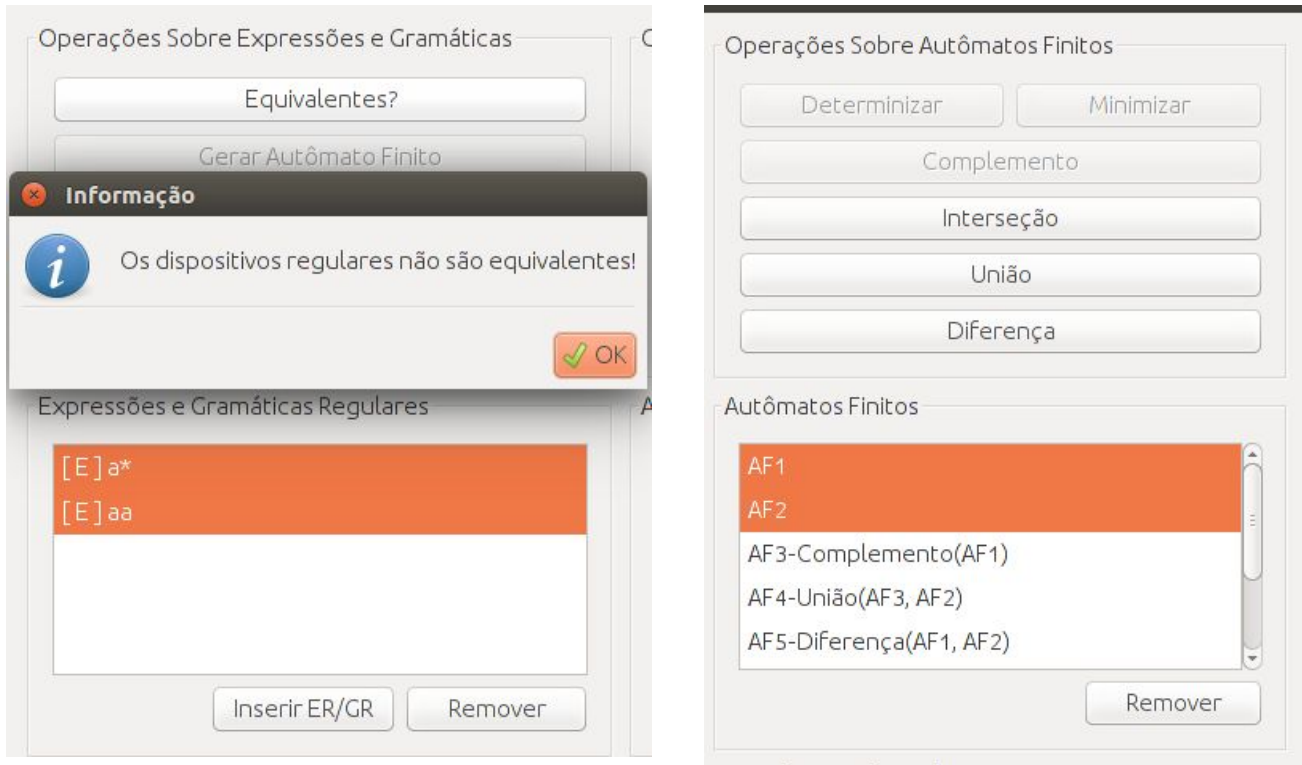
- Tipo do Dispositivo:** ☐ Expressão Regular ☒ Gramática Regular
- Descrição Textual:** A text area containing the grammar rule `S->aA|b|`.
- Buttons:** "Cancelar" and "Ok".

Após a inserção da nova Expressão Regular ou Gramática Regular, elas irão surgir no campo “Expressões e Gramáticas Regulares”. Caso o usuário queira criar o Autômato Finito respectivo a alguma delas, basta selecionar a ER/GR desejada, e clicar no botão “Gerar Autômato Finito”. O novo Autômato será adicionado ao campo “Autômatos Finitos”, e caso o usuário queira visualizá-lo, deve-se clicar no mesmo. O resultado de todas essas interações pode ser visto na próxima imagem:



Também é possível realizar outras operações sobre as ER/GR e sobre os autômatos. Caso o usuário queira verificar se dois autômatos são equivalentes, ele deve segurar a tecla CTRL e clicar em duas opções na lista de ER/GR. Caso queira fazer a intersecção, união ou diferença entre dois autômatos, deve-se segurar a tecla CTRL e clicar nos dois autômatos desejados. As operações de intersecção e diferença irão gerar novos autômatos intermediários, mostrando as operações formais utilizadas.

As duas imagens a seguir ilustram o que foi dito no último parágrafo.



3. O que faltou

Infelizmente, não foram implementadas as funções de: reverso de LR's (parte do item 5), fechamento de GR's (parte do item 6) e reconhecimento de sentenças e enumeração (item 7).

4. Documentação

A documentação de cada método está presente no próprio código.

4.1 Principais métodos

- public void determinize() - presente na classe FiniteAutomaton, efetua a operação de determinização do Autômato caso o mesmo não esteja determinizado.
- public void minimize() - presente na classe FiniteAutomaton, efetua a operação de minimização do Autômato. Invoca os métodos presentes na classe FiniteAutomaton:
 - determinize(): descrição acima
 - removeDeadStates(): remove estados mortos

- `removeUnreachableStates()` : remove estados inalcançáveis
- `complete()`: completa o automato caso necessário
- `public FiniteAutomaton complement()` - presente na classe `FiniteAutomaton`, realiza o complemento do Autômato.
- `public List<FiniteAutomaton> unionView(FiniteAutomaton automaton)` - presente na classe `FiniteAutomaton`, realiza a união dos Autômatos.
- `public List<FiniteAutomaton> intersection(FiniteAutomaton automaton)` - presente na classe `FiniteAutomaton`, realiza a intersecção dos Autômatos.
- `public List<FiniteAutomaton> difference(FiniteAutomaton automaton)` - presente na classe `FiniteAutomaton`, realiza a diferença entre Autômatos.
- `public FiniteAutomaton buildAutomaton()` - presente na classe `RegularExpressionAutomatonBuilder`, transforma a entrada em um Autômato.