

## Manipulação de Linguagens Regulares e Linguagens Livres de Contexto

Alunos: Arthur Mesquita Pickcius

Francisco Luiz Vicenzi

João Fellipe Uller

### Objetivo do Trabalho

O objetivo deste trabalho é a implementação dos algoritmos relacionados a manipulação de Linguagens Regulares e Livres de Contexto. Tais algoritmos são úteis na implementação de Geradores de Analisadores Léxicos e Sintáticos ou dos próprios analisadores.

### Definição do Trabalho

Elaborar uma aplicação, com interface gráfica para facilitar a iteração, para manipular Autômatos Finitos, Gramáticas Regulares, Expressões Regulares e Gramáticas Livres de Contexto e Autômatos de Pilha. A aplicação deve suportar:

- (✓) Leitura, gravação e edição de AF, GR e ER.
- (✓) Conversão de AFND (com e sem  $\epsilon$ ) para AFD.
- (✓) Conversão de AFD para GR e de GR para AFND.
- (✓) Reconhecimento de sentenças em AF.
- (e) Minimização de AFD.
- (f) União e interseção de AFD.
- (g) Conversão de ER para AFD (usando o algoritmo baseado em árvore sintática - Livro Aho - seção 3.9).
- (h) Leitura, gravação e edição de GLC.
- (i) Transformação de GLC para uma GLC na forma normal de Chomsky.
- (j) Eliminação de recursão a esquerda
- (k) Fatoração
- (l) Reconhecimento de sentenças em AP (teorema  $GLC \leftrightarrow AP$ )

### Formato de Entrega

Dia 02/05 - entrega da parte 1 - itens (a), (b), (c) e (d) da Definição do Trabalho

Dia 30/05 - entrega da parte 2 - itens (e), (f), (g) e (h) da Definição do Trabalho

Dia 20/06 - entrega da parte 3 - itens (i), (j), (k) e (l) da Definição do Trabalho

### Ferramentas utilizadas

O presente trabalho foi realizado na linguagem Python versão 3.7, utilizando, para a interface gráfica, a biblioteca Qt. É possível instalá-la utilizando o seguinte comando: `pip install PySide2`

### Para rodar o programa

Rodar o arquivo `main.py` com o comando `python3 main.py`

### Modelagem e Estrutura de Dados

O programa é orientado a objetos, em que cada modelo estudado (AF, GR, ER) consiste em uma classe. Para construir as transições e produções, utilizamos dicionários e listas.

### Utilização e exemplos

O programa apresenta interface bem intuitiva, permitindo a leitura, gravação e edição de AFs, GRs e ERs. Sendo assim, é possível manipular uma instância por vez.

Os exemplos apresentados foram retirados das listas de exercícios, nomeados no formato <L:lista\_E:exercicio\_T:letra>. Para manipulação, basta carregar o arquivo desejado, disponível na pasta examples, pelo menu File->Open.

### Operações de conversão

As operações de determinização e conversão podem ser encontradas no menu Convert nas janelas dos autômatos, gramáticas e expressões regulares.

### Salvar/abrir arquivos

As operações de abrir e salvar podem ser encontradas no menu File nas janelas dos autômatos, gramáticas e expressões regulares.

### Autômatos Finitos

Para **criar um novo automato** basta ir no menu File->new, e aparecerá uma janela perguntando os símbolos do alfabeto. Os símbolos devem ser inseridos separados por vírgulas E.g. a , b , c . Os estados e transições podem ser inseridos utilizando os botões da barra lateral.

Para fazer o **reconhecimento de uma sentença**, existe o menu Input com as opções de Fast Run e Step by State . . . (esta última disponível apenas na visualização pelo terminal) para uma sentença e a opção Multiple Run para testar um conjunto de sentenças.

### Gramáticas Regulares

Para **editar uma gramática** é necessário clicar em algum item da coluna ls e adicionar um símbolo não terminal. Com o símbolo não terminal na coluna ls, a coluna rs é liberada para que sejam adicionadas as regras de produção separadas por | , seguindo as regras das gramáticas regulares. E.g. S -> aA|a .

### Expressões Regulares

A **edição de expressões regulares** é feita de forma simples, onde o usuário apenas digita a expressão. E.g. abba\*abba(a|b)\*