

# Trabalho I: Algoritmos para Manipulação de Linguagens Regulares

Linguagens Formais e Compiladores  
Prof<sup>a</sup>. Jerusa Marchi

## 1. Objetivo do Trabalho:

O objetivo deste trabalho é o desenvolvimento dos algoritmos básicos para tratamento de linguagens regulares. Trata-se da implementação das transformações entre mecanismos geradores e reconhecedores. Tais algoritmos serão utilizados no trabalho II (desenvolvimento de um analisador Léxico).

## 2. Definição do Trabalho:

Devem ser implementados os seguintes algoritmos:

- (a) Transformação de AFND para AFD
- (b) Transformação de AFND com  $\varepsilon$ -transições para AFD
- (c) Transformação de Gramáticas Regulares para Autômatos Finitos e de Autômatos Finitos para Gramáticas Regulares
- (d) Conversão de Expressões Regulares para Autômatos Finitos e de Autômatos Finitos para Expressões Regulares

## 3. Realização:

O trabalho deverá ser realizado por grupos de no máximo 3 estudantes. As entradas e saídas devem ser via arquivo texto. Sugiro utilizar as notações habituais de AF, Gramáticas e ER para elaborar as entradas e saídas. A linguagem é escolha do grupo.

## 4. Formato e Data da Entrega:

Deve ser entregue no dia 02/10 e defendido na semana de 05 a 09/10.

A entrega deve ser feita pelo moodle em um arquivo <Nome1Nome2Nome3>.zip que quando descompactado gere um diretório <Nome1Nome2Nome3> e dois subdiretório 2: Algoritmos e Testes (este quesito é fundamental!). Os arquivos de teste devem ser na forma <teste1AfndAfd.in> e <teste2AfndAfd.out> respectivamente. Deve haver dentro do diretório principal um arquivo README.txt contendo cabeçalho (instituição, departamento, nomes dos integrantes do grupo, data) e informações sobre os fontes (linguagem utilizada, detalhes para execução do programa, etc.), entradas e saídas de cada algoritmo. Cada algoritmo deve ter seu fonte separado em um arquivo nomeado de acordo com a sua funcionalidade (Se a implementação for orientada a objetos, então o nome do método deve ser a funcionalidade do algoritmo). Bibliotecas auxiliares podem ser utilizadas visando limpar o código (para fazer entrada e saída de dados, por exemplo). Estes arquivos (todos) também devem ter cabeçalho e devem ser bem comentados.

**IMPORTANTE!** Trabalhos fora do padrão não serão corrigidos.