

# LFA - ALGORITMOS DE CONVERSÃO DE AUTÔMATOS

Ana Paula da Silva Cunha - Matrícula : 0011252 <sup>1</sup>

,Suená Batista Galoneti - Matrícula : 0011251 <sup>2</sup>

<sup>1</sup>Instituto Federal de Minas Gerais, Formiga, MG

**Resumo:** Este documento apresenta o relatório do desenvolvimento do primeiro trabalho da disciplina que objetivava a implementação de alguns algoritmos.

Palavras-Chaves: LFA; Algoritmos.

## 1 INTRODUÇÃO

O conceito de autômato o define como um conjunto finito de estados.

Autômatos também podem ser definidos através de uma quintupla : Conjunto de Estados, Alfabeto, Função de Transição, Estado Inicial(se determinístico, e Conjunto de Estados Iniciais se não determinístico) e Conjunto de Estados Finais.

Descrevendo seu funcionamento ou caminharmento vemos que um autômato parte de um estado inicial e caminha para um próximo estado de acordo com a função de transição e a palavra que está sendo reconhecida, e está deve conter apenas elementos do alfabeto do autômato para ser aceita.

### 1.1 EQUIVALÊNCIA

A equivalência de autômatos se dá a partir da equivalência de seus estados, sendo o principal o estado inicial, pois partindo de uma mesma entrada o comportamento dos dois autômatos deve ser o mesmo para que estes sejam equivalentes.

### 1.2 MINIMIZAÇÃO

Um autômato é possível de minimização se este possuir estados equivalentes, ou seja, se dois estados pertencentes ao autômato desempenharem o mesmo papel, se este caso ocorrer o autômato mínimo irá apresentar apenas um desses estados.

### 1.3 MULTIPLICAÇÃO

Utilizamos a multiplicação de autômatos quando, de maneira bruta, quisermos transformar dois autômatos em um. Ou seja, partindo de dois autômato distintos e pré definidos, realizarmos uma operação que ao final resulte em um único autômato. Essas operações podem ser :

- Complemento

- Diferença
- Interseção
- União

## 2 DESENVOLVIMENTO

Foi optado pelo desenvolvimento dos algoritmos acima descritos na linguagem Java.

Para a montagem dos autômatos e saída gráfica está sendo utilizado o JFlap.

Inicialmente foram criadas duas classes nomeadas como, a primeira, de AFD e, a segunda, FunçãoTransição.

Dentro da classe AFD foi implementado o método para leitura (utilizando árvore) do arquivo XML.

Outro método presente nesta classe é o para a obtenção do complemento, aonde verifica quais estados são finais e quais não são e transforma os estados não finais nos novos estados finais.

Para a implementação da união, interseção e diferença de autômatos fez necessário a implementação da multiplicação. A multiplicação consistiu em unir os alfabetos dos dois autômatos em um novo alfabeto e fazer um novo caminhamento verificando os nós visitados e os a visitar e utilizando da função transição obtendo novos estados para o autômato final(o autômato final terá seus estados finais definidos de acordo com a operação escolhida) e estes estados também devem passar pela verificação de visitados ou a visitar e serem caminhados.

Os estados serão marcados com finais de acordo com a operação : se for união os estados marcados como finais serão as combinações dos estados finais originais, em outras palavras, os estados novos começados com o estado final do primeiro autômato e os novos estados terminados com o estado final do segundo autômato. No caso da interseção serão finais apenas os novos estados que forem compostos exclusivamente por estados finais dos autômatos originais. A diferença nada mais é do que E NÃO, sendo assim ela irá aceitar como estados finais novos apenas os estados começados pelo estados finais do primeiro autômato e terminado pelos estados não finais do segundo autômato.

Na implementação ainda encontram-se outros pequenos métodos com sub tarefas utilizadas para a implementação das operações do trabalho.

## Referências

- [1] Ferreira, B.  
Introdução a linguagem Java  
Programação Orientada a Objetos  
Prof. Dr. Bruno Ferreira