

Apresentação Trabalho 2 de LFA

Conversão AFN em AFD

Conteúdo da apresentação:

- Descrição geral do algoritmo.
- Estruturas de dados utilizadas para representar a quintupla.
- AFNs testados durante a implementação.
- O que é apresentado como saída do algoritmo.

Descrição Geral do Algoritmo

O algoritmo funciona a base de **verificação de estados**, começando com o inicial, e é um **loop incremental**, ou seja, a medida que vamos descobrindo novos estados dentro da conversão, os colocamos em análise.

Descrição Geral do algoritmo

O algoritmo verifica os estados e depois utiliza o alfabeto para buscar nas transições as passagens de estados para armazenar o estado novo que vai ser criado e o conjunto de estados.

Descrição Geral do algoritmo

No momento em que estes estados são encontrados, é feita uma série de verificações pra **adicionar os estados à lista** e logo depois **adicionar às transições**.

Então, dentro do código são criado **novos estados, estados de transição e os estados finais** para a quintupla do **AFD**

Estruturas utilizadas para representar a quíntupla

- **Estados** = ['q0','q1',...,'qn']
- **Alfabeto** = ['a1','a2',...,'an']
- Os **Estados de Transição** serão representados pela Matriz de Estados:
['a1','q0','q1'], ['a2','q1','q2'], ['a2','q1','q2'], ..., ['an','qn','qn']
- **Estado inicial** = 'q0'
- **Conjunto de Estados Finais** = ['q1','q2',...,'qn']

AFNs testados durante a implementação

- **L1 = AFN que tem 00 ou 11 como subcadeia.**
- **L2 = AFN que tem aa ou bb como subpalavra.**
- **L3= AFN que termina com 000.**
- **L4 = AFN que termina com aaa.**
- **L5 =AFN que termina com 01.**
- L6 = AFN que termina com ab**

Saída do algoritmo

- A saída do algoritmo é um **arquivo .txt** com a quintupla que será **entrada para o AFD.**