

Trabalho Final

1 Definição

Em grupos de 3 **alunos**, realizar as seguintes tarefas, divididas em 2 partes:

1. Cenário Real

- (a) Selecionar e descrever um **cenário de um sistema real** (linguagem) que contenha, pelo menos, 5 operações/ações/eventos relevantes;
- (b) **Definir formalmente a linguagem** L_c que descreve o comportamento do sistema escolhido, associando símbolos (alfabeto) às suas operações e apresentando as regras de formação de palavras (execuções válidas);
- (c) Apresentar um **AFN** M_c que reconheça a linguagem L_c ;
- (d) Criar uma **lista de 10 palavras** sobre o alfabeto de L_c , sendo que, destas, 5 devem pertencer à $ACEITA(M_c)$ e 5, à $REJEITA(M_c)$;

2. Implementação

Criar um **programa** que, dado um autômato finito não determinístico (AFN) M que reconhece uma linguagem regular L , execute as seguintes operações:

- (a) Construa o **autômato finito determinístico** M_D de M usando o algoritmo visto em aula;
- (b) Dada uma palavra w , **indique se** $w \in ACEITA(M_D)$. Em caso positivo, o programa deve apresentar o caminho (sequência alternada de estados visitados e símbolos lidos) que leva do estado inicial até o estado de aceitação. Caso w seja rejeitada, o programa deve indicar se o reconhecimento de w terminou em um estado não final ou aconteceu uma situação de função indefinida;
- (c) Apresentar o resultado da avaliação das propriedades de M_D em relação à linguagem vazia e finita/infinita.

2 Instruções

- O programa do item 2 pode ser implementado usando qualquer linguagem de programação, desde que o código fonte seja bem documentado;
- O **formato do arquivo de entrada** contendo a definição do AFD deve seguir o seguinte padrão:

```

<M>=<ini>, {<finais>}
<q_0>
<s_0>:<q_1>
...
<s_n>:<q_n>
<q_1>
<s_0>:<q_1>
...
<s_n>:<q_n>
...

```

onde:

< M >: nome dado ao autômato;

< ini >: indica o estado inicial do autômato, sendo que *ini* é um estado do autômato;

< finais >: indica o conjunto de estados finais do autômato;

< q_i >: para $0 \leq i \leq n$, com $n \in \mathbb{N}$ e $n \geq 0$, representa um estado qualquer do autômato;

< s_i >: para $1 \leq i \leq n$, com $n \in \mathbb{N}$ e $n \geq 1$, representa um símbolo qualquer do alfabeto da linguagem reconhecida pelo autômato;

< s_i >:< q_j >: descreve a função programa aplicada a partir de um estado qualquer q_i e um símbolo de entrada s_i que leva a computação a um estado q_j .

Exemplo:

```

AUTÔMATO= (q0, {q1, q3})
q0
a:q1
b:q2
q1
b:q2
a:q0
a:q1
q2
a:q3
b:q2
q3
a:q3
b:q2

```

- Todas as operações do programa devem seguir os algoritmos vistos em aula para garantir o resultado correto. **Qualquer otimização ou alteração deve ser devidamente documentada e associada a uma argumentação de correção;**
- Todas as entradas devem ser fornecidas **via teclado (palavra de entrada) ou seleção de arquivo (AFN);**

1. **Descrição do cenário**, das operações e da formalização da linguagem (itens 1a e 1b);
2. Arquivo contendo o **AFN do item 1c no formato de entrada** definido.
3. Arquivo do JFLAP ou arquivo de imagem contendo o **AFN** do item 1c;
4. Arquivo com a **lista de palavras** do item 1d;
5. **Fontes e executável do programa** descrito no item 2 da Seção 1 (bem como qualquer biblioteca/arquivo/instrução adicional necessário para execução do programa).