

ICMC-USP
Trabalho em Grupo 1
SCC-0505

1º. Semestre de 2022

Professor: João Luís G.Rosa - e-mail: joaoluis@icmc.usp.br

Estagiário PAE: Fernando S. de Aguiar Neto - e-mail: fernando.soares.aguiar@usp.br

1 Objetivo

Desenvolver o entendimento de Linguagens Formais e seu potencial de representação através da implementação de simuladores de autômatos finitos.

2 Descrição

O trabalho deve ser preferencialmente realizado em grupos de no máximo cinco alunos. Cada grupo deve projetar e desenvolver a aplicação abaixo, empregando qualquer linguagem de programação.

- *Simulador Universal de Autômatos Finitos*: O programa deve aceitar a especificação de um AFD ou AFN e a partir daí para uma dada lista de cadeias, dizer quais as que pertencem (saída: **aceita**) e quais as que não pertencem (saída: **rejeita**) à linguagem reconhecida pelo autômato. *Notem que seguiremos a definição dada em aula de permitir apenas um estado inicial, mesmo nas AFN.*

3 Produto

O programa a ser implementado neste projeto deve seguir rigorosamente os formatos de entrada e saída (ver seção “Arquivos Texto de Entrada e de Saída” abaixo), e enviado ao Escaninho de um membro do grupo na plataforma Tidia-Ae 4.0 (ae4.tidia-ae.usp.br), na página da disciplina SCC-0505, até às 23h59 do dia **06 de maio de 2022**. **O prazo final é improrrogável.** Além do programa fonte e um executável em Windows, um relatório com a descrição do trabalho deverá ser entregue (ver seção “Critérios” abaixo).

4 Critérios

Os critérios de correção dos trabalhos são:

1. (80%) **Implementação**: O programa funciona corretamente para todos os casos de teste;
2. (20%) **Documentação**: Relatório simples que explica as técnicas utilizadas para implementar a máquina escolhida. Discutir a qualidade da solução implementada, a estruturação do código e a eficiência da solução em termos de espaço e tempo. A documentação deverá ser submetida ao Tidia-Ae, juntamente com o código fonte e um

arquivo executável em Windows. **IMPORTANTE:** Incluir explicações claras sobre como executar o programa em um arquivo texto de nome **manualT1.txt**. Portanto, quatro arquivos devem ser anexados: o relatório (arquivo txt ou PDF), o arquivo fonte (compilável), executável e manual.

Atenção: O plágio (cópia) de programas não será tolerado. Quaisquer programas similares terão nota zero independente de qual for o original e qual for a cópia.

5 Arquivos Texto de Entrada e de Saída

Dois arquivos: ExemploIn.txt e ExemploOut.txt estão disponíveis para conferência.

Arquivo Texto de Entrada:

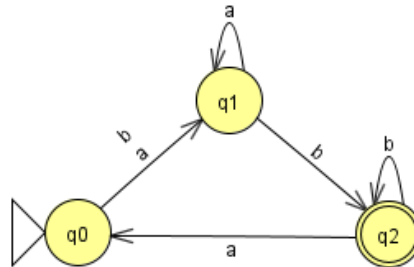
- 1ª. Linha: número de estados: para o conjunto de estados Q , assume-se os nomes dos estados de q_0 a q_{n-1} , onde n é o número de estados (Obs.: q_0 é o estado inicial, sempre). Assuma $1 \leq n \leq 10$;
- 2ª. Linha: o conjunto de símbolos terminais (Σ): entrar com a quantidade de símbolos terminais seguida dos elementos separados por espaço simples. Assume-se tamanho máximo igual a 10;
- 3ª. Linha: o conjunto de estados de aceitação (F): entrar com a quantidade de estados de aceitação seguida dos elementos separados por espaços. Lembre-se de entrar apenas com os números de 0 a 9;
- 4ª. Linha: o número de transições (δ) da máquina (máximo de 50).
- a partir da 5ª Linha: as transições: entra-se com um δ em cada linha, com os elementos separados por espaço: $q \ x \ q'$, onde $q, q' \in Q$, $x \in \Sigma \cup \{\lambda\}$. Represente a cadeia vazia (λ) como “-”.
- Linha depois das transições: entrar com o número de cadeias de entrada (máximo de 10).
- Próximas Linhas: cadeias de entrada: entrar com uma em cada linha. Comprimento máximo de cada cadeia = 20 símbolos.

Arquivo Texto de Saída:

- a partir da 1ª. Linha: a informação sobre a aceitação ou não da respectiva cadeia de entrada, **na ordem** do arquivo de entrada. Se a cadeia de entrada pertencer à linguagem reconhecida pelo autômato, a cadeia de saída será “aceita”. Caso a cadeia de entrada não pertença à linguagem reconhecida pelo autômato, a cadeia de saída será “rejeita”.

6 Exemplo

- Autômato finito determinístico (AFD) que processa a linguagem regular $(a+b)a^*bb^*(a(a+b)a^*bb^*)^*$.



Arquivo Texto de Entrada¹:

1. 3
2. 2 a b
3. 1 2
4. 6
5. 0 a 1
6. 0 b 1
7. 1 a 1
8. 1 b 2
9. 2 a 0
10. 2 b 2
11. 10
12. abbbba
13. aabbbb
14. bbabbabbabb
15. bbbbbbbbbbb
16. -
17. abababababab
18. bbbbaabbbb
19. abba
20. a
21. aaa

¹Os números das linhas **não** devem aparecer no arquivo-texto. Estão colocados aqui apenas para facilitar o entendimento.

Arquivo Texto de Saída:

1. rejeita
2. aceita
3. aceita
4. aceita
5. rejeita
6. rejeita
7. aceita
8. rejeita
9. rejeita
10. rejeita