



Universidade de Itaúna
Curso de Ciência da Computação
Disciplina: Teoria da Computação
Professor: Felipe Cunha

Trabalho Prático

Data de Entrega: a definir

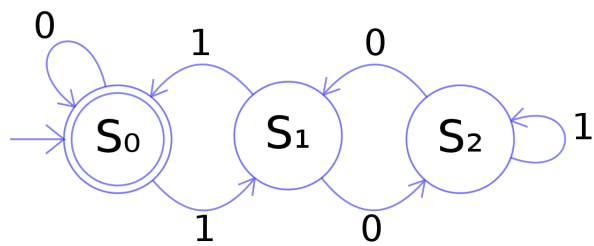
Neste trabalho o aluno deverá implementar um simulador de autômatos finitos determinísticos – *AFD*, que permita ao usuário testar o reconhecimento de diversas cadeias para qualquer *AFD* de entrada. Assim, o programa desenvolvido pelo aluno deverá permitir que o mesmo introduza qualquer *AFD*. Desta forma, para montar o autômato $A = (E, \Sigma, \delta, I, F)$ o usuário deverá informar:

- Os estados do *AFD*
- O alfabeto do *AFD*
- As funções de transições para os símbolos
- O estado inicial
- Os estados de finais/aceitação

Para simplificar, permita a restrição do alfabeto a apenas 2 símbolos. O programa deverá rejeitar autômatos inválidos (Não determinísticos). Os componentes do autômato poderão ser lidos diretamente pelo console ou por um arquivo de entrada, ficando a critério do programador a implementação que achar mais conveniente. Após o registro de todos os componentes do *AFD*, deverá ser permitido ao usuário a verificação se uma palavra pertence ou não à linguagem definida pelo autômato.

Informações Adicionais:

- O trabalho deverá ser em grupos de máximo 3 alunos.
- A escolha da linguagem para a implementação do programa ficará a critério de cada aluno.
- O trabalho deverá ser apresentado em horário de aula, onde o aluno deverá entregar o código fonte e um manual de uso do simulador.
- A fórmula para desconto por atraso na entrega do trabalho prático é: $\frac{2^{d-1}}{0.32}\%$ onde d é o atraso em dias úteis. Entregas após a meia-noite da data especificada serão invariavelmente penalizadas, mesmo se realizadas no próximo dia útil. Note que após 5 dias úteis, o trabalho não pode ser mais entregue.



PADRÃO DE ENTRADA DOS DADOS:

3 2 6 1 1 # $A = (E, \Sigma, \delta, I, F)$

S0 S1 S2 # estados

0 1 # Σ {alfabeto da linguagem}

S0 # {estado inicial}

S0 # {estados finais}

S0 S0 0

S0 S1 1

S1 S2 0

S1 S0 1

S2 S1 0

S2 S2 1

STRINGS PARA TESTE:

0000000 => RECONHECE

1001 => RECONHECE

1010 => NÃO RECONHECE

11111 => NÃO RECONHECE