

SCC-ICMC-USP  
**Trabalho em Grupo 2**  
SCC-0505

1º. Semestre de 2010  
Professor: João Luís G. Rosa - e-mail: [joaoluis@icmc.usp.br](mailto:joaoluis@icmc.usp.br)  
versão 1 - 17/5/2010

## 1 Objetivo

Desenvolver o entendimento de Linguagens Formais e seu potencial de representação através da implementação de processadores de linguagens formais.

## 2 Descrição

O trabalho deve ser preferencialmente realizado em grupos de três. Cada grupo deve projetar e desenvolver um *Simulador Universal de Máquinas de Turing*, empregando uma das seguintes linguagens de programação: C ou C++:

- *Simulador Universal de Máquinas de Turing*: O programa deve aceitar a especificação de uma máquina de Turing 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 pela máquina.

## 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), uma vez que todos os projetos serão submetidos, no período de **17 e 18 de junho de 2010**, ao corretor automático Boca (<http://blacklabel.intermedia.icmc.usp.br/boca/scc0505/>). Recomenda-se que a primeira submissão ocorra antes do prazo final, para que sejam possíveis eventuais correções. **O prazo final é improrrogável.** Além do programa, 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%) O programa funciona corretamente para todos os casos de teste;
2. (20%) **Documentação**: relatório simples que explica as técnicas utilizadas para implementar a máquina. 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 entregue na primeira aula após o final do prazo de submissão do trabalho, ou seja, no dia 24/6/2010.

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.

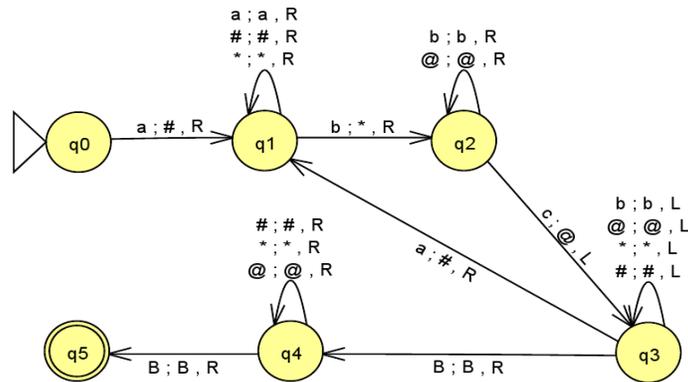
## 5 Arquivos Texto de Entrada e de Saída

### Arquivo Texto de Entrada:

- 1<sup>a</sup>. 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). Portanto, basta entrar com o número de estados. Assuma  $1 \leq n \leq 10$ ;
- 2<sup>a</sup>. Linha: o conjunto de símbolos terminais ( $\Sigma$ ): entrar com a quantidade de símbolos terminais seguida dos elementos separados por espaço simples. Assume-se tamanho máximo igual a 10;
- 3<sup>a</sup>. Linha: a quantidade de símbolos de  $\Sigma'$  (alfabeto estendido de fita) não presentes em  $\Sigma$ , seguido pelos símbolos, separados por espaço simples;
- 4<sup>a</sup>. Linha: o estado de aceitação: entrar com o estado de aceitação ( $q_a$ ). Lembre-se de entrar apenas com os números de 0 a 9;
- 5<sup>a</sup>. Linha: o número de transições ( $\delta$ ) da máquina (máximo de 50).
- a partir da 6<sup>a</sup> Linha: as transições: entra-se com um  $\delta$  em cada linha, com os elementos separados por espaço:  $q x q' y D$ , onde  $q, q' \in Q$ ,  $x, y \in \Sigma'$  e  $D \in \{R, L, S\}$ . Assuma os limites da fita como o símbolo branco (B). Represente a cadeia vazia ( $\lambda$ ) 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 1<sup>a</sup>. 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 Exemplos

- Exemplo: Máquina de Turing determinística que processa a linguagem  $a^n b^n c^n$ , com  $n > 0$ .

Arquivo Texto de Entrada<sup>1</sup>:

1. 6
2. 3 a b c
3. 4 # \* @ B
4. 5
5. 18
6. 0 a 1 # R
7. 1 # 1 # R
8. 1 a 1 a R
9. 1 \* 1 \* R
10. 1 b 2 \* R
11. 2 b 2 b R
12. 2 @ 2 @ R
13. 2 c 3 @ L
14. 3 # 3 # L
15. 3 \* 3 \* L
16. 3 @ 3 @ L
17. 3 b 3 b L
18. 3 a 1 # R
19. 3 B 4 B R
20. 4 # 4 # R

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

21. 4 \* 4 \* R
22. 4 @ 4 @ R
23. 4 B 5 B R
24. 10
25. abbcca
26. aabbcc
27. bac
28. aaabbbcccc
29. -
30. abcabc
31. abc
32. abcc
33. c
34. aaabbbbccc

**Arquivo Texto de Saída:**

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

## 7 Notas

1. Apenas máquinas determinísticas serão testadas.
2. Apenas linguagens recursivas serão testadas, ou seja, linguagens para as quais a máquina de Turing sempre para independentemente da aceitação,
3. O processamento da cadeia começa no estado  $q_0$  e no símbolo mais à esquerda de  $w$ . Exemplo: para a fita  $abba$ , a descrição instantânea inicial é  $q_0abba$ .