

INE 5415 - Teoria da Computação

Trabalho 1: Máquinas de Turing

Fernando Paladini, Fristtram Helder Fernandes

Enunciado

1) Implemente Máquinas de Turing com fita única para computar as seguintes linguagens:

- a. $L = \{ w\#w \mid w \in \{a,b,c\}^* \}$
- b. $L = \{ \#x_1\#x_2\#\dots\#x_n \mid x_i \in \{0,1\}^* \text{ e } x_i \neq x_j \text{ para cada } i \neq j \}$

2) Implemente Máquinas de Turing Multifitas para computar as seguintes linguagens:

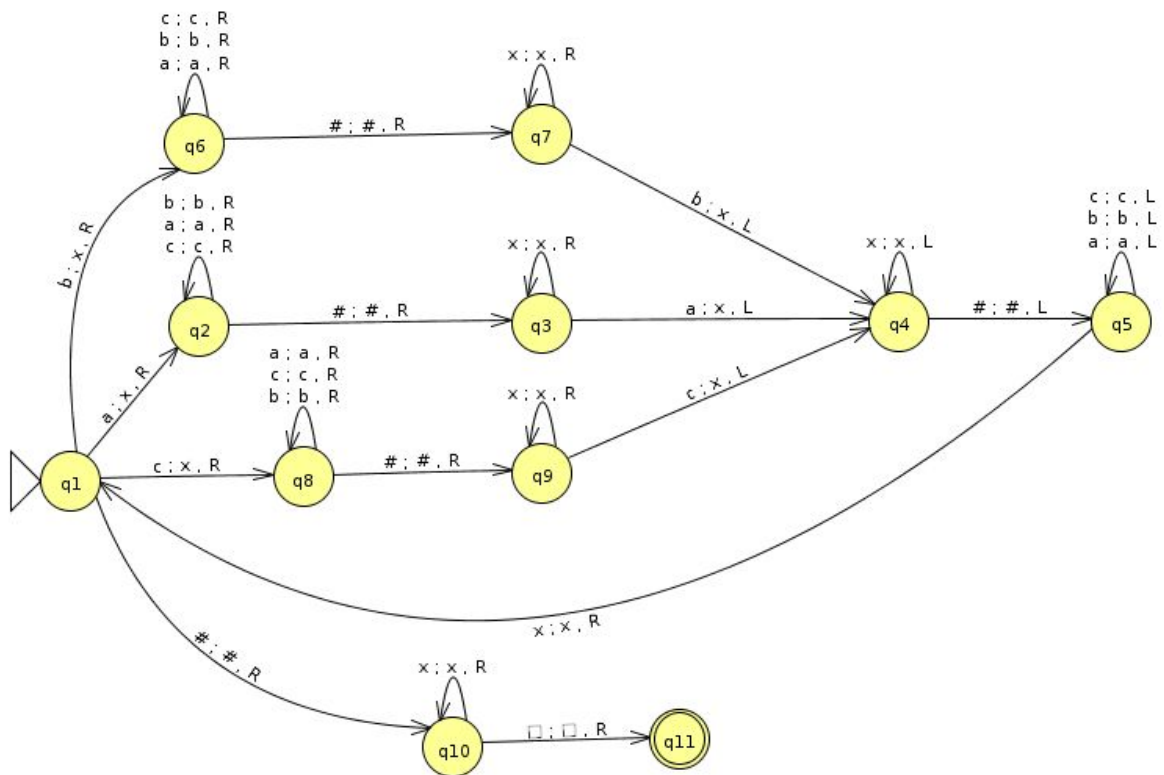
- a. $L = \{ ww \mid w \in \{0,1\}^* \}$
- b. $L = \{ w(w^r)w \mid w \in \{a,b\}^* \}$
- c. Um somador binário

Questão 1, exercício A.

$L = \{ w\#w \mid w \in \{a,b,c\}^* \}$

Descrição do funcionamento em alto nível: A máquina aceita entradas vazias (" # "), pois não há nenhuma descrição na linguagem que restrinja palavras vazias separadas pelo delimitador #. Nesse caso, uma transição é realizada para q10 e depois para q11, que é um estado aceitador. Para outras entradas, a máquina vai para um "estado específico" de acordo com o caracter lido na entrada (a, b ou c). O caracter lido é escrito como "x" na fita. Agora a máquina começa a ler (e escrever na fita) toda a entrada até encontrar o delimitador "#", quando então avança o cabeçote para a direita uma última vez. Caso existam marcações "x" após o delimitador, a máquina percorre todas elas até encontrar o mesmo caracter que iniciou o "estado específico". Caso não encontre esse caracter, vai para o estado morto; caso encontre, escreve "x" na fita e avança o cabeçote para a esquerda. No estado q4, a máquina começa a voltar para o começo da fita, passando por todos os caracteres já marcados com "x" após o marcador. Quando a máquina chegar no delimitador "#" (estado q5), ela avança mais uma posição para a esquerda e continua a percorrer a fita para a esquerda (passando pelas entradas a, b ou c) até encontrar um "x". Quando isso acontece a máquina avança o cabeçote para a direita e agora está pronta novamente para terminar de processar a entrada (atualmente se encontra no estado q1). Caso esteja no estado q1 e não encontre mais caracteres para processar, a máquina vai encontrar o delimitador "#" e então ir para o estado q10, onde percorrerá a fita até chegar no seu final e então ir para o estado aceitador q11.

Codificação:



Testes:

JFLAP : (exercício 1a.jff)

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

Input	Result
#	Accept
aa#aa	Accept
abc#abc	Accept
bbbb#bbbb	Accept
braceyourselves#starwars7iscoming	Reject
abd#abd	Reject
ccc#cc	Reject

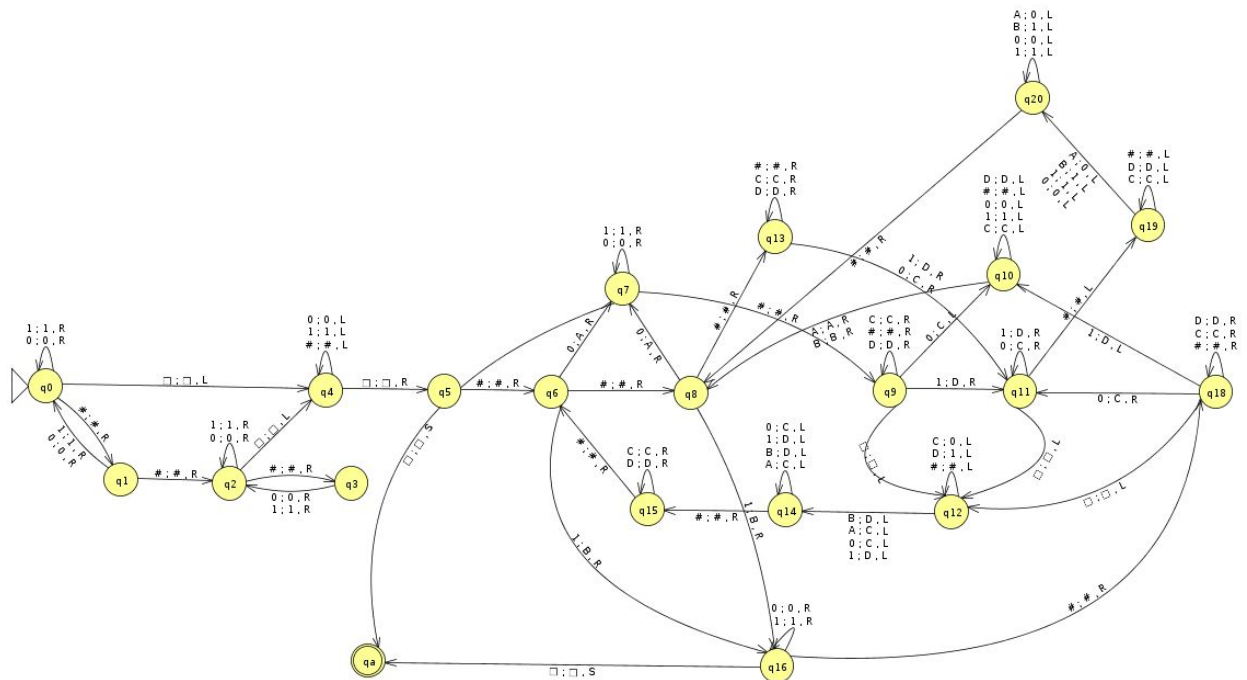
Load Inputs Run Inputs Clear Enter Lambda View Trace

Questão 1, exercício B.

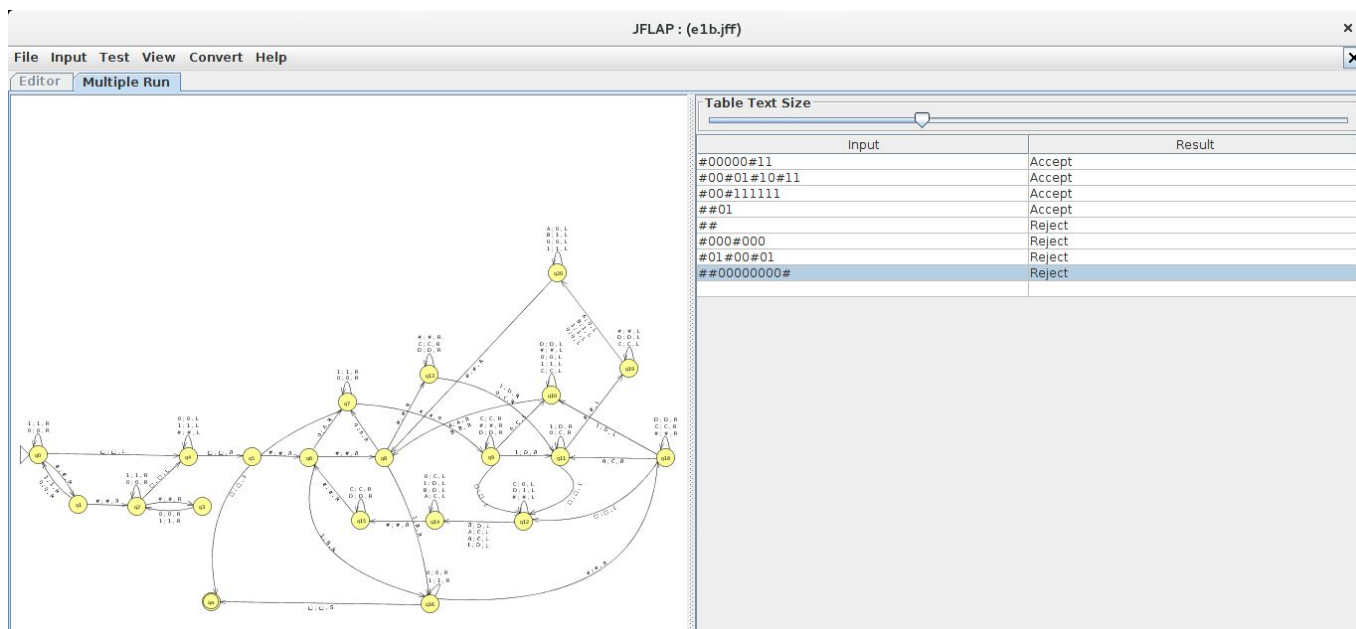
$L = \{ \#x_1\#x_2\#\dots\#x_n \mid x_i \in \{0,1\}^* \text{ e } x_i \neq x_j \text{ para todo } i \neq j \}$

Descrição do funcionamento em alto nível: A máquina compara x_i e x_j para todo i diferente de j (com exceção à palavra vazia, que é garantida no início do procedimento). Para a comparação, um x_i é fixado e comparado com x_{i+1} , x_{i+2} , x_{i+3} , ..., x_n . Depois do final da comparação, uma nova “palavra” (sub-palavra talvez seja o termo mais adequado) é fixada (x_{i+1}) e uma nova comparação com as “sub-palavras” posteriores, separadas por “#”, é realizada. Quando o caracter “#” não é mais encontrada, significa que a comparação chegou ao fim.

Codificação:



Testes:

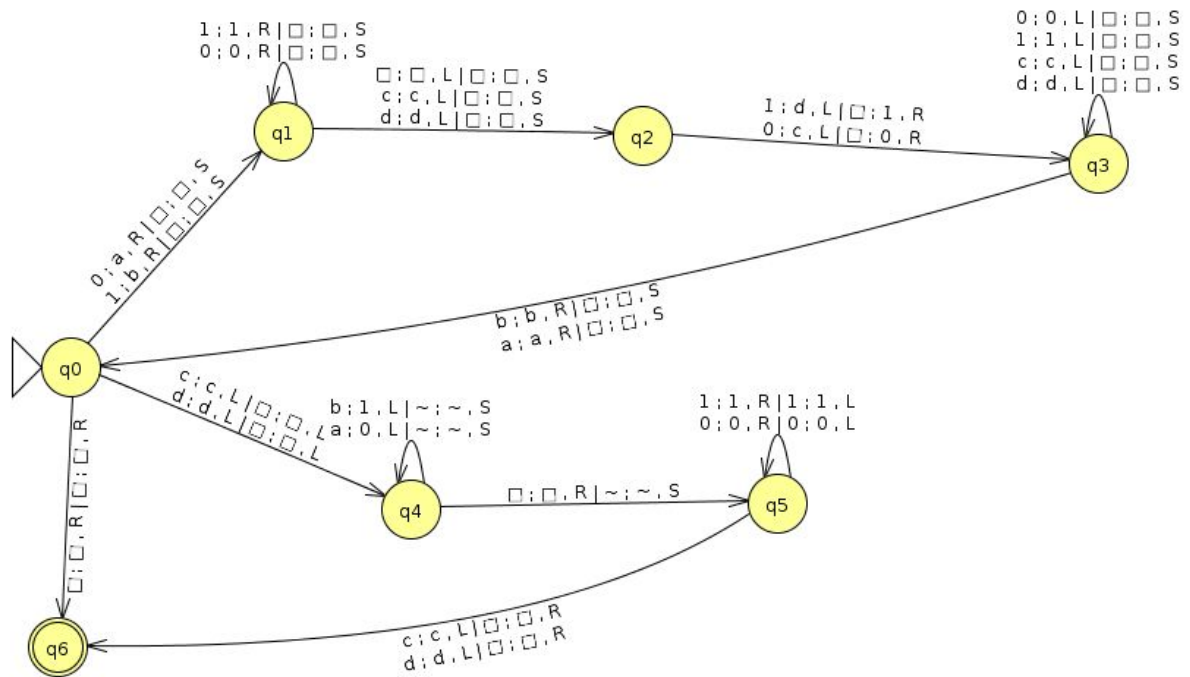


Questão 2, exercício A.

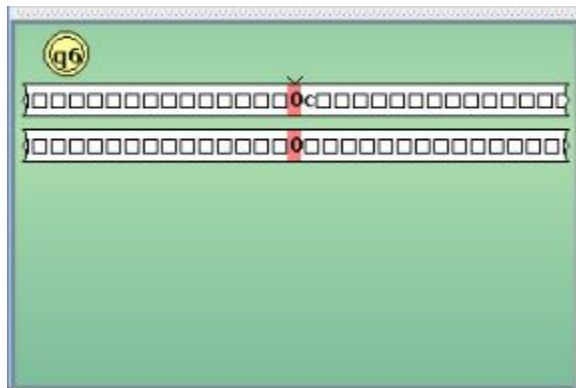
$$L = \{ ww \mid w \in \{0,1\}^* \}$$

Descrição do funcionamento em alto nível: A máquina lê a entrada, então os primeiros caracteres são marcados com “a” (0) e “b” (1) e os últimos são marcados com “c” (0) e “d” (1). Quando os caracteres da última palavra (considerando, nesse caso, a última palavra como a última parte da palavra “ww”) são marcados, a fita 2 também vai sendo escrita com os mesmos caracteres. Quando todos os zeros e uns da palavra “ww” estiverem marcados, a primeira palavra (apenas “w”, não “ww”) é remarcada com 0’s e 1’s. Depois disso, a fita 1 é comparada indo do “começo” da fita para a direita, até chegar no fim da primeira palavra, enquanto que a fita 2 é comparada indo do “final” da fita para a esquerda, até chegar no começo da segunda palavra; se alguém caracter divergir, significa que a entrada é inválida, caso contrário a entrada deve ser aceita.

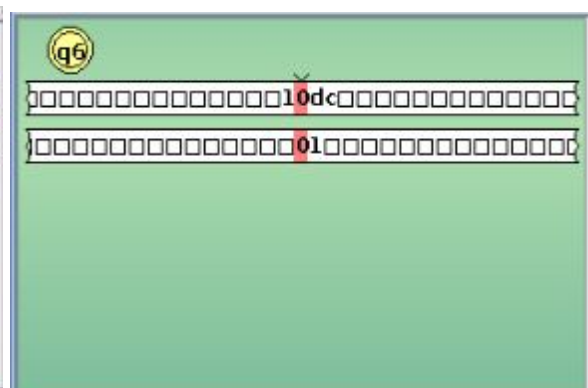
Codificação:



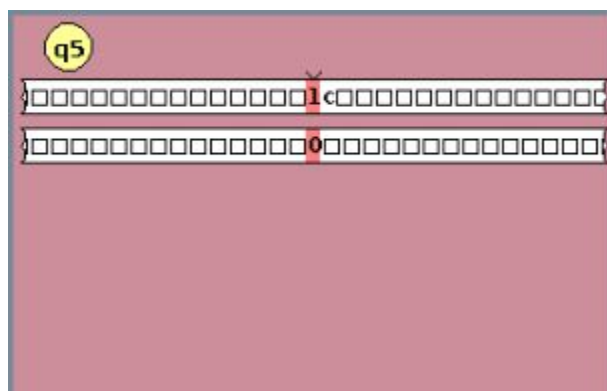
Teste:



Teste bem sucedido com entrada "00".



Teste bem sucedido com entrada "1010".



Teste mal sucedido com entrada "10".



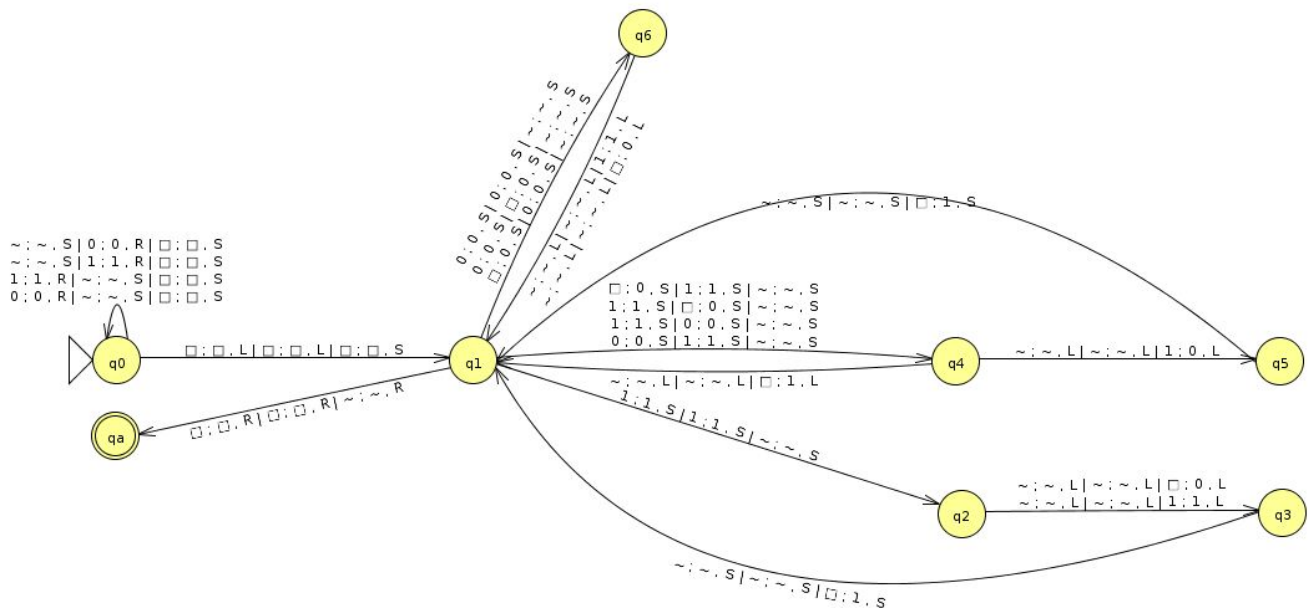
Teste mal sucedido com entrada "100101".

Obs: não conseguimos realizar os testes no “Multiple Run”, justamente por isso mostramos a execução dos testes em “Step”. Por causa da maior dificuldade para gerar os testes, fizemos apenas para 2 casos de cada tipo.

Questão 2, exercício C.

Descrição do funcionamento em alto nível: A máquina aceita como entradas valores binários, a partir dos qual é possível obter como resultado a soma dos valores que foram dados como entrada. Para tal, é necessário inserir os valores a serem somados no Input 1 (Fita 0) e no Input 2 (Fita 1). A entrada de cada fita é analisada individualmente e ambas as fitas são apontadas para o final das entradas binárias. Após isso vamos ao estado “q1”, onde é analisado o último bit das duas fitas (o último bit ainda não calculado) e então as propriedades de soma binária (já conhecidas) são aplicadas na 3ª fita, que após realizar todas as iterações conterá o valor resultante da soma.

Codificação:



Testes:

File
Input
Test
View
Convert
Help

Editor
Multiple Run

Table Text Size

Input 1	Input 2	Input 3	Result
01	01		Accept
10	10		Accept
01	00		Accept
00	00		Accept
-10			Reject
21	12		Reject
33	33		Reject
5	4		Reject

Obs: embora não seja possível ver o resultado das somas no modo “Multiple Run”, se executar cada teste individualmente o resultado da soma será sempre o esperado.