Universidade Federal de Santa Catarina - UFSC
Departamento de Informática e Estatística (INE)
Ciência da Computação
Disciplina INE5415 - Teoria da Computação
Prof. Jerusa Marchi

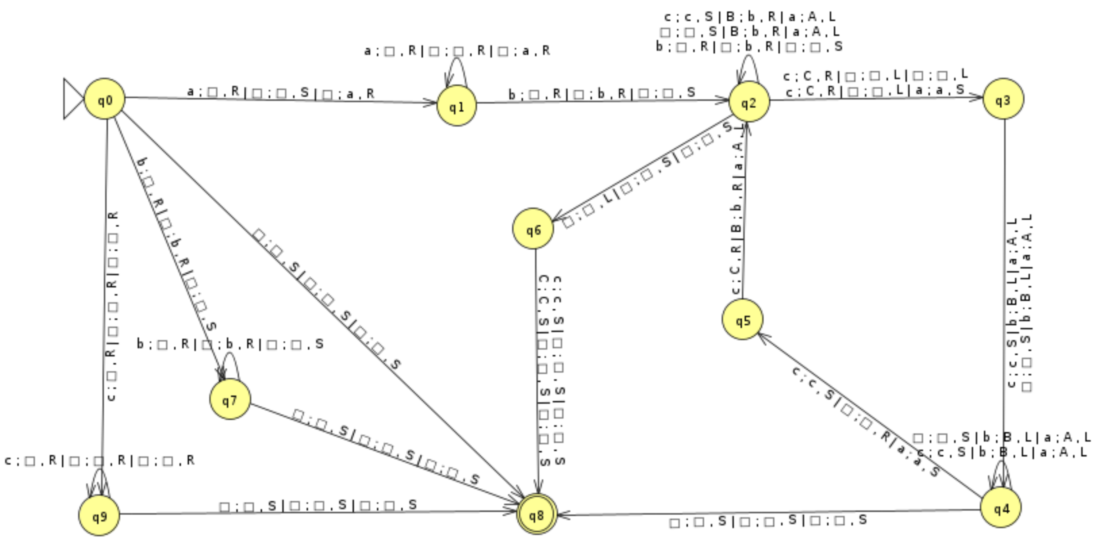
Lívia Corazza Ferrão (21202119)
Brenda Silva Machado (21101954)

Trabalho 1 - Máquinas de Turing

1) Máquina 1a - Algoritmo para uma máquina de 3 fitas de forma que o custo computacional seja n^2. L = {aibjck | i, j, k ∈ N e i = j × k}.

1) Copiar os a's para a terceira fita. O(i)
2) Copiar os b's para a segunda fita. O(j)
3) Mantêm-se os c's na primeira fita. O(1)
4) Para cada c ímpar, a fita 2 desloca-se para a esquerda e marca os b's com B's e para cada c par, a fita 2 desloca-se para a direita e marca B's com b's. A fita 1 com os c's desloca-se para a direita. O(j * k)
5) Para cada b ou B marcado, marca-se um a com A e desloca-se a terceira fita para a esquerda. O(j)
6) Palavra vazia vai direto para o estado de aceitação. O(1)
7) Se a palavra começa com b, percorre-se a fita até o fim e verifica se não existem c's em seguida, então aceita. O(n)
8) Se a palavra começa com c, percorre-se a fita até o fim e verifica se só existem c's, então aceita. O(n)

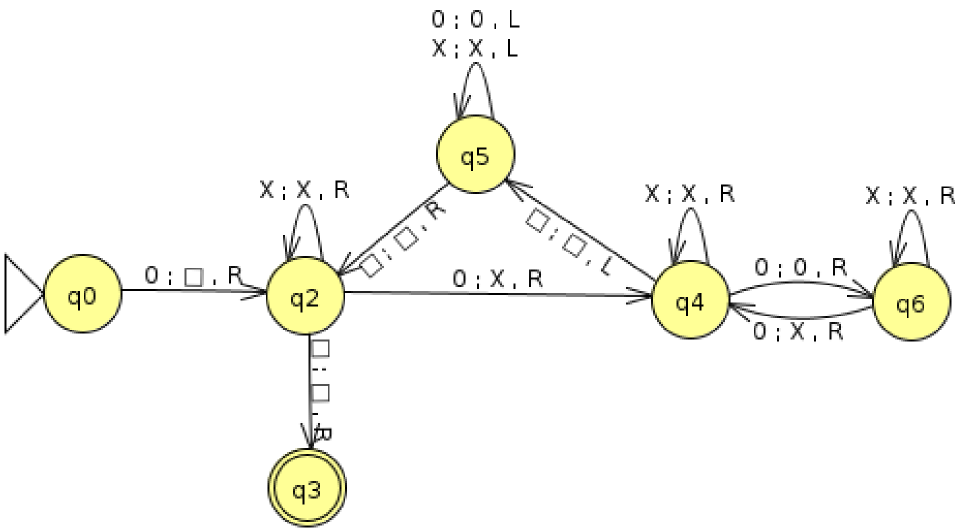
A fita 1 é percorrida 1 vez, a fita 2 é percorrida c vezes e a fita 3 é 1 vez (sempre à esquerda). No total teremos c*b como complexidade. (j * k).



2) Máquina 1b - Implementação idêntica a do livro do Sipser, com complexidade n lg n. L = {02n | n ≥ 0}

1) Percorre-se a fita da esquerda para a direita. O(log n)
   1.1) Marca os 0's de forma alternada. O(n)
2) Se no estágio 1, a fita continha um único 0, aceite. O(n)
3) Se no estágio 1, a fita continha mais que um único 0, o número de 0's era ímpar, rejeite. O(n)
4) Retorne a cabeça para a extremidade esquerda da fita. O(n)
5) Volte ao estágio 1. O(1)

A implementação não contém o estado de rejeição. A cada iteração o número de 0's é reduzido pela metade (log n). E percorre-se a fita inteira (n). A complexidade final é n log n.



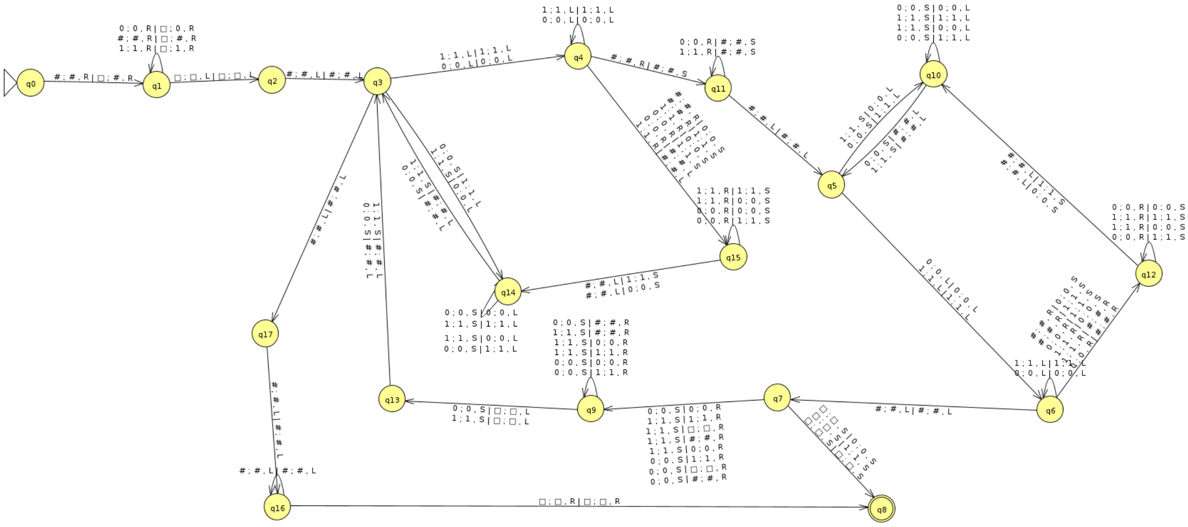
3) Máquina 2b - L = {#x1#x2#...#xn# | xi ∈ {0, 1}∗ e ∀xi∃xj tal que xi = xj para algum i 6 = j}.

Obs. Sendo n o tamanho da entrada e m o número de palavras x.

a) **Avalie a sua máquina original.**

   1) Copia a fita de entrada para a segunda fita. O(n)
   2) A cada palavra da fita compara-se com cada palavra da outra fita. O(n*m)
   3) Volte para o início da fita a cada nova comparação. O(n*m)

Por fim, a complexidade da máquina é O(n*m), uma vez que essa é a operação com maior complexidade.



b) Implementação trabalho 2.

Para i de 1 até n, faça:

1) Copia a palavra wi da fita 1 para a fita 2. O(1)
2) Apaga a palavra wi da fita 1. O(1)
3) Compara todas as outras palavras da fita 1 com wi, marcando aquelas que são iguais com um caractere especial. O(n-i)

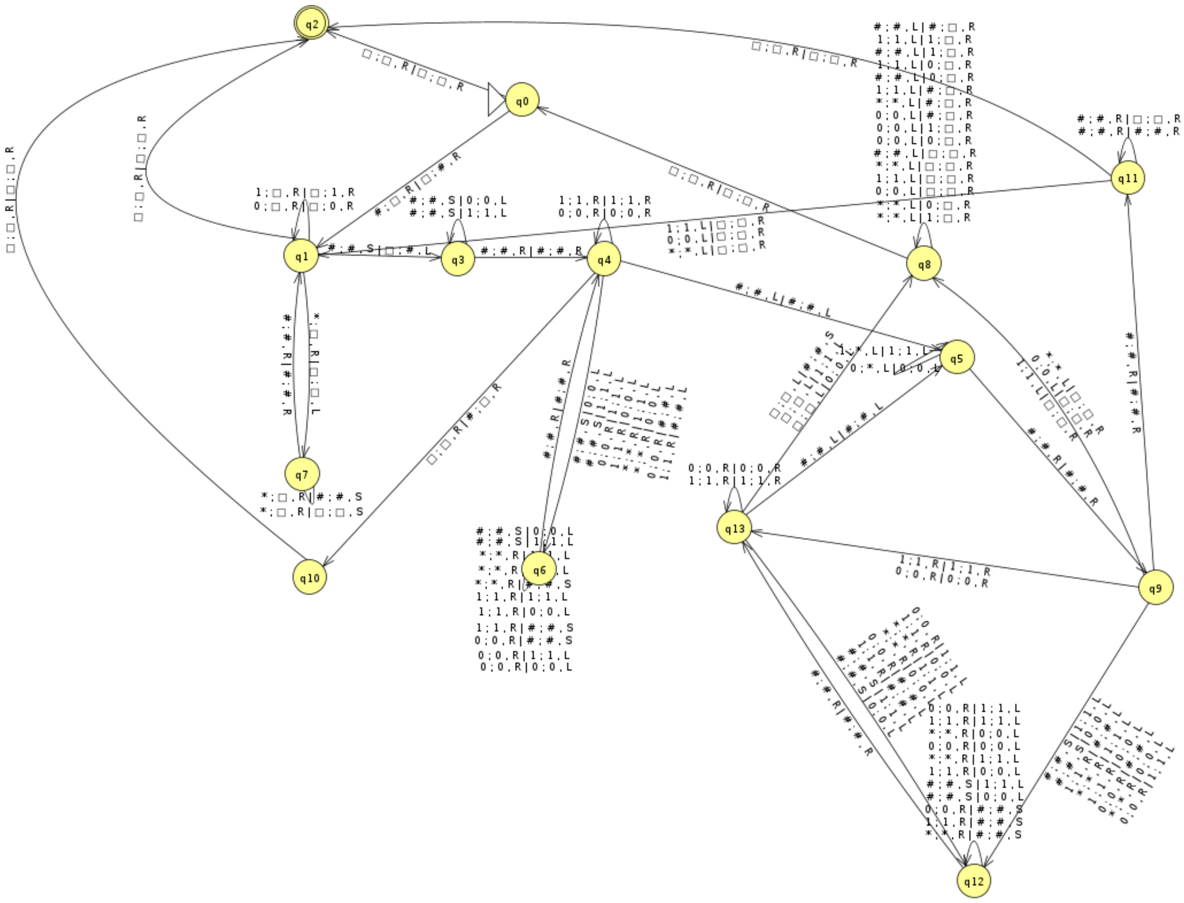
A diminuição de n ocorre porque há uma eliminação contínua do número de palavras, logo a cada loop a operação de comparação executa em tempo menor, logo com uma complexidade menor. Assim, o somatório fica na forma n + (n-1) + (n-2) + ... + 1, o que resulta numa complexidade de $O(n^2)$.

Essa complexidade pode ser obtida utilizando a fórmula da soma:

$$S = ((n-1) * n) / 2 \Rightarrow (n^2 - n) / 2 \Rightarrow (n^2 / 2 - n / 2); = O(n^2/2)$$

Como podemos cortar o 1/2, a complexidade final fica:

$$O(n^2)$$


q0
q1
q2
q3
q4
q5
q6
q7
q8
q9
q10
q11
q12
q13