Teoria da computação

Problem set 3

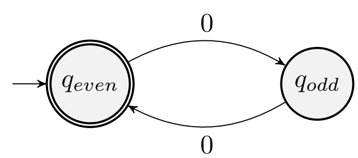
Rodrigo Santos

Universidade NOVA de Lisboa

Exercício 1

Para cada uma das linguagens abaixo descreva um AFD que a reconhece através do seu diagrama de estados e de uma definição formal

(a) $L = \{0^{2n} \mid n \in \mathbb{N}\}$



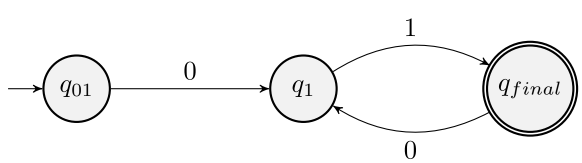
A descrição formal do AFD é:

$$M_1 = (\{q_{even}, q_{odd}\}, \{0\}, \delta, q_{even}, \{q_{even}\})$$

onde δ é representado da seguinte maneira:

δ	*0*
q_{even}	q_{odd}
q_{odd}	q_{even}

(b) $L = \{(01)^n \mid n \in \mathbb{N}\}$



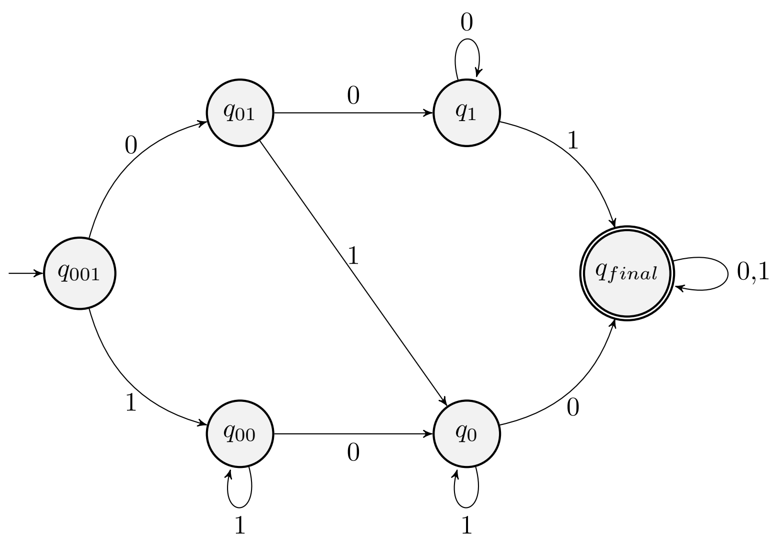
A descrição formal do AFD é:

$$M_2 = (\{q_{01}, q_1, q_{final}\}, \{0, 1\}, \delta, q_{01}, \{q_{final}\})$$

onde δ é representado da seguinte maneira:

δ	*0*	*1*
q_{01}	q_1	$\perp$
q_1	$\perp$	q_{final}
q_{final}	q_1	$\perp$

(c) A linguagem L das strings sobre $\{0, 1\}$ que contêm pelos menos dois $0's$ e pelo menos um 1.



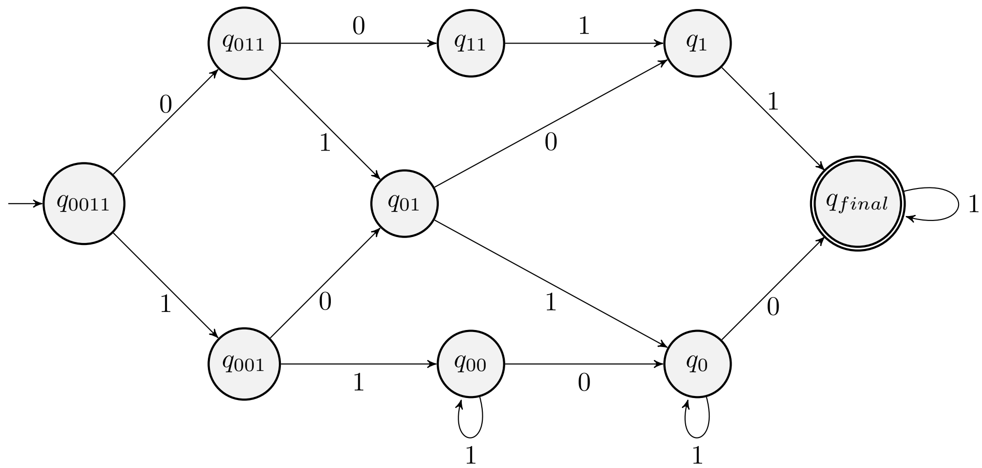
A descrição formal do AFD é:

$$M_3 = (\{q_{001}, q_{01}, q_{00}, q_1, q_0, q_{final}\}, \{0, 1\}, \delta, q_{001}, \{q_{final}\})$$

onde δ é representado da seguinte maneira:

δ	*0*	*1*
q_{001}	q_{01}	q_{00}
q_{00}	q_0	q_{00}
q_{01}	q_1	q_0
q_1	q_1	q_{final}
q_0	q_{final}	q_0
q_{final}	q_{final}	q_{final}

(d) A linguagem L das strings sobre $\{0, 1\}$ que contêm exatamente dois $0's$ e pelo menos dois $1's$.



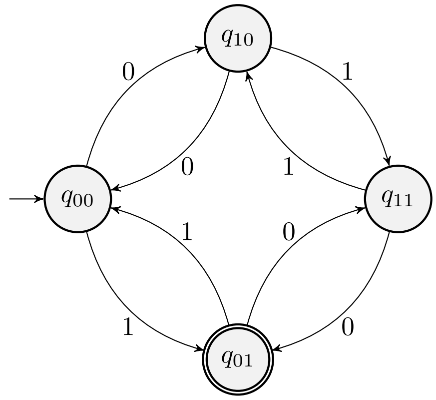
A descrição formal do AFD é:

$$M_4 = (\{q_{0011}, q_{011}, q_{001}, q_{11}, q_{01}, q_{00}, q_1, q_0, q_{final}\}, \{0, 1\}, \delta, q_{0011}, \{q_{final}\})$$

onde δ é representado da seguinte maneira:

δ	*0*	*1*
q_{0011}	q_{011}	q_{001}
q_{001}	q_{01}	q_{00}
q_{011}	q_{11}	q_{01}
q_{01}	q_0	q_1
q_{00}	q_0	q_{00}
q_{11}	$\perp$	q_1
q_0	q_{final}	q_0
q_1	$\perp$	q_{final}
q_{final}	$\perp$	q_{final}

(e) A linguagem L das strings sobre $\{0,1\}$ com um número par de $0's$ e um número ímpar de $1's$.



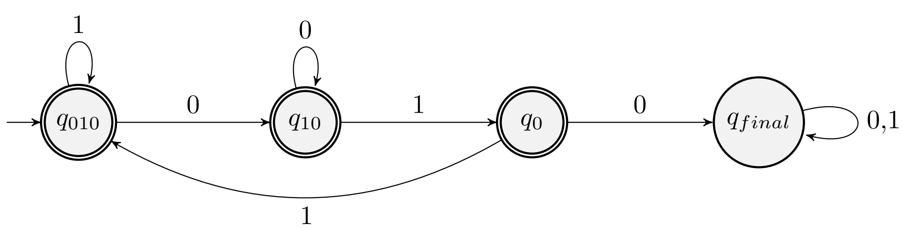
A descrição formal do AFD é:

$$M_5 = (\{q_{00}, q_{10}, q_{01}, q_{11}\}, \{0,1\}, \delta, q_{00}, \{q_{01}\})$$

onde δ é representado da seguinte maneira¡:

δ	*0*	*1*
q_{00}	q_{10}	q_{01}
q_{10}	q_{00}	q_{11}
q_{01}	q_{11}	q_{00}
q_{11}	q_{01}	q_{10}

(f) A linguagem L das strings sobre $\{0,1\}$ que não contêm a substring 010.



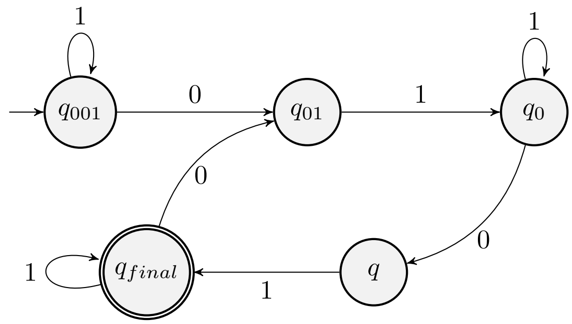
A descrição formal do AFD é:

$$M_6 = (\{q_{010}, q_{10}, q_0, q_{final}\}, \{0,1\}, \delta, q_{010}, \{q_{010}, q_{10}, q_0\})$$

onde δ é representado da seguinte maneira:

δ	*0*	*1*
q_{010}	q_{10}	q_{010}
q_{10}	q_{10}	q_{0}
q_{0}	q_{final}	q_{010}
q_{final}	q_{final}	q_{final}

(g) A linguagem L das strings sobre $\{0,1\}$ com um número par de $0's$ e em que cada 0 é sempre seguido de pelo menos um 1.



A descrição formal do AFD é:

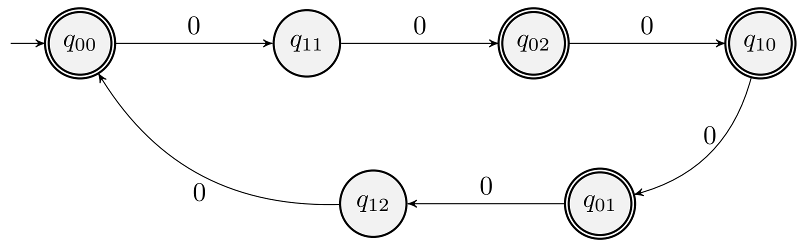
$$M_7 = (\{q_{001}, q_{01}, q_0, q, q_{final}\}, \{0,1\}, \delta, q_{001}, \{q_{final}\})$$

onde δ é representado da seguinte maneira:

δ	*0*	*1*
q_{001}	q_{01}	q_{001}
q_{01}	$\perp$	q_{0}
q_{0}	q	q_{0}
q	$\perp$	q_{final}
q_{final}	q_{01}	q_{final}

(h) A linguagem L das strings sobre $\{0\}$ com tamanho divisível por 2 ou por 3.

q_{ij} onde $i = n\%2$ e $j = n\%3$.



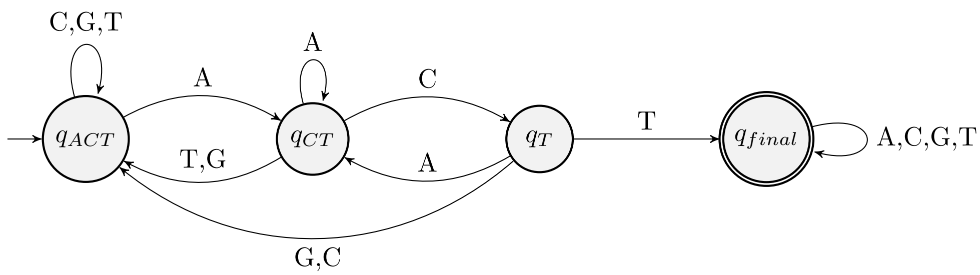
A descrição formal do AFD é:

$$M_8 = (\{q_{00}, q_{11}, q_{02}, q_{10}, q_{01}, q_{12}\}, \{0\}, \delta, q_{00}, \{q_{00}, q_{02}, q_{10}, q_{01}\})$$

onde δ é representado da seguinte maneira:

δ	*0*
q_{00}	q_{11}
q_{11}	q_{02}
q_{02}	q_{10}
q_{10}	q_{01}
q_{01}	q_{12}
q_{12}	q_{00}

(i) A linguagem L das strings sobre $\{A, C, G, T\}$ que contêm pelo menos uma ocorrência da substring ACT.



A descrição formal do AFD é:

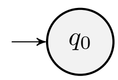
$$M_9 = (\{q_{ACT}, q_{CT}, q_T, q_{final}\}, \{A, C, G, T\}, \delta, q_{ACT}, \{q_{final}\})$$

onde δ é representado da seguinte maneira:

δ	A	C	G	T
q_{ACT}	q_{CT}	q_{ACT}	q_{ACT}	q_{ACT}
q_{CT}	q_{CT}	q_T	q_{ACT}	q_{ACT}
q_T	q_{CT}	q_{ACT}	q_{ACT}	q_{final}
q_{final}	q_{final}	q_{final}	q_{final}	q_{final}

(j) $L = \varnothing$

Iremos assumir Σ como $\{0, 1\}$.



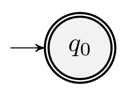
A descrição formal do AFD é:

$$M_{10} = (\{q_0\}, \{0, 1\}, \delta, q_0, \varnothing)$$

onde δ é representado da seguinte maneira: $\delta : S \times \Sigma \to S$ dada por $\delta(q_0, 0) = \perp$ e $\delta(q_0, 1) = \perp$

(k) $L = \varepsilon$

Iremos assumir Σ como $\{0, 1\}$.

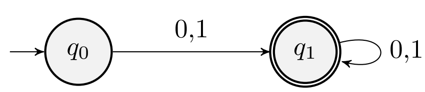


A descrição formal do AFD é:

$$M_{11} = (\{q_0\}, \{0, 1\}, \delta, q_0, \{q_0\})$$

onde δ é representado da seguinte maneira: $\delta : S \times \Sigma \to S$ dada por $\delta(q_0, 0) = \perp$ e $\delta(q_0, 1) = \perp$

(l) $L = \{0, 1\}^* \setminus \{\varepsilon\}$



A descrição formal do AFD é:

$$M_{12} = (\{q_0, q_1\}, \{0, 1\}, \delta, q_0, \{q_1\})$$

onde δ é representado da seguinte maneira:

δ	0	1
q_0	q_1	q_1
q_1	q_1	q_1

Exercício 2

Para cada um dos $AFD's$ que construiu nas alíneas (a) a (g) do Exercício 1, descreva a sequência de estados percorridos no input 0100110 e diga se esta string é aceite ou não.

(a)

$\delta(q_{even}, 0100110) = \delta(\delta(q_{even}, 0), 1001110) = \delta(\delta(q_{odd}, 1), 001110)$, $\delta(q_{odd}, 1) = \perp$. logo esta sequência não é aceite.

(b)

$\delta(q_{01}, 0100110) = \delta(\delta(q_{01}, 0), 100110) = \delta(\delta(q_1, 1), 00110) = \delta(\delta(q_{final}, 0), 0110) = delta(\delta(q_1, 0), 110)$ como $\delta(q_1, 0) = \perp$, a sequência não é aceite.

(c)

$\delta(q_{001}, 0100110) = \delta(\delta(q_{001}, 0), 100110) = \delta(\delta(q_{01}, 1), 00110) = \delta(\delta(q_0, 0), 0110) = \delta(\delta(q_{final}, 0), 110) = \delta(\delta(q_{final}, 1), 10) = \delta(\delta(q_{final}, 1), 0) = \delta(q_{final}, 0) = q_{final}$, como $q_{final} \in F$, a sequência é aceite.

(d)

$\delta(q_{0011}, 0100110) = \delta(\delta(q_{0011}, 0), 100110) = \delta(\delta(q_{011}, 1), 00110) = \delta(\delta(q_{01}, 0), 0110) = \delta(\delta(q_1, 0), 110)$, como $\delta(q_1, 0) = \perp$, a sequência não é aceite.

(e)

$\delta(q_{00}, 0100110) = \delta(\delta(q_{00}, 0), 100110) = \delta(\delta(q_{10}, 1), 00110) = \delta(\delta(q_{11}, 0), 0110) = \delta(\delta(q_{01}, 0), 110) = \delta(\delta(q_{11}, 1), 10) = \delta(\delta(q_{10}, 1), 0) = \delta(q_{11}, 0) = q_{01}$, como $q_{01} \in F$, a sequência é aceite.

(f)

$\delta(q_{010}, 0100110) = \delta(\delta(q_{010}, 0), 100110) = \delta(\delta(q_{10}, 1), 00110) = \delta(\delta(q_0, 0), 0110) = \delta(\delta(q_{final}, 0), 110) = \delta(\delta(q_{final}, 1), 10) = \delta(\delta(q_{final}, 1), 0) = \delta(q_{final}, 1) = q_{final}$, como $q_{final} \notin F$, a sequência não é aceite.

(g)

$\delta(q_{001}, 0100110) = \delta(\delta(q_{001}, 0), 100110) = \delta(\delta(q_{01}, 1), 00110) = \delta(\delta(q_0, 0), 0110) = \delta(\delta(q, 0), 110)$, como $\delta(q, 0) = \perp$, a sequência não é aceite.

Exercício 3

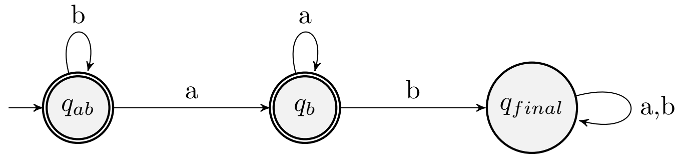
Seja L uma linguagem regular. Quando é que temos $\epsilon \in L$?

Se L é regular quer dizer que existe um AFD $M = (S, \Sigma, \delta, s, F)$ com função de transição total que reconhece L. Para M reconhecer ϵ, basta termos $s \in F$. Ou seja o estado inicial também ser estado de aceitação.

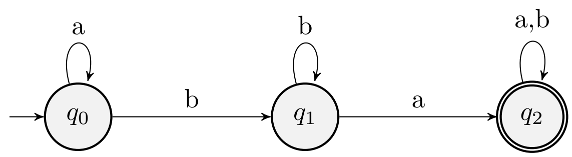
Exercício 4

Para cada uma das linguagens L abaixo descreva um AFD que a reconhece através do seu diagrama de estados. Sugestão: Primeiro construa um AFD que reconhece o complemento $\overline{L}$ e depois converta-o para um AFD que reconhece L.

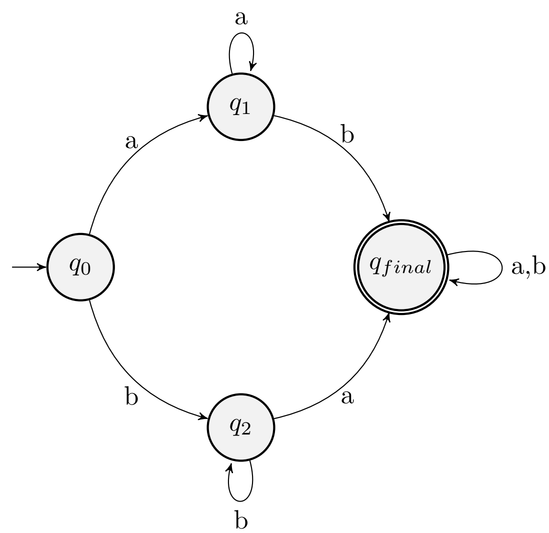
(a) A linguagem L sobre $\{a, b\}$ cujas strings não contêm a substring ab.



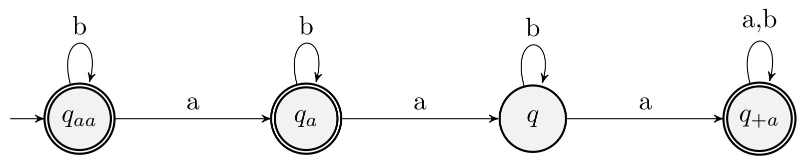
(b) $L = \{a, b\}^* \setminus \{a^m b^n \mid m, n \in \mathbb{N}\}$



(c) $L = \{a, b\}^* \setminus (\{a\}^* \cup \{b\}^*)$



(d) A linguagem L sobre $\{a, b\}$ cujas strings não contêm exatamente dois $a's$.



Exercício 5

Sejam L_1 e L_2 linguagens regulares sobre o mesmo alfabeto Σ. Mostre que $L_1 \cap L_2$ também é regular.

Como L_1 e L_2 são regulares, existem $AFD's$ $M_1 = (S_1, \Sigma, \delta_1, s_1, F_1)$ e $M_2 = (S_2, \Sigma, \delta_2, s_2, F_2)$, com funções de transição δ_1 e δ_2 totais tal que M_1 e M_2 aceitam L_1 e L_2 respetivamente. Queremos construir um AFD M a partir de M_1 e M_2 que reconhece $L_1 \cap L_2$. Dado um input w, queremos simular as computações de M_1 e M_2 em w lado-a-lado, e aceitar w exatamente quando ambos M_1 e M_2 aceitam w.

Para implementar a estratégia acima vamos definir um novo AFD $M = (S, \Sigma, \delta, s, F)$. Para tal necessitamos de a cada momento saber os estados atuais de M_1 e M_2 na computação. Ou seja os estados de M podem ser descritos como *pares* de estados de M_1 e M_2, em que cada par representam o estado atual dos dois $AFD's$. Portanto queremos $S = S_1 \times S_2$. Se M estiver no estado $(q_1, q_2) \in S_1 \times S_2$ num dado ponto da computação e ler o símbolo $a \in \Sigma$, queremos simular a computação de M_1 e M_2,

para tal deveríamos atualizar q_1 para $\delta_1(q_1, a)$ e q_2 para $\delta_2(q_2, a)$. Definimos então a função de transição δ como
$$\delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a)).$$
Para o estado inicial de M, queremos escolher o par de estados iniciais de M_1 e M_2. Escrevemos então $s = (s_1, s_2)$. Resta especificar F, o conjunto de estados finais de M. Queremos aceitar w se ambas as computação de M_1 e M_2 acabarem em estados de aceitação. Para tal queremos escolher F como o conjunto de pares $(q_1, q_2) \in S$ tal que $q_1 \in F_1$ e $q_2 \in F_2$. Ou seja definimos o conjunto de estados de aceitação F como
$$F = (F_1 \times F_2)$$
Agora que definimos M formalmente, queremos provar que $L(M) = L_1 \cap L_2$.Para tal vamos mostrar $L(M) \subseteq L_1 \cap L_2$ e $L_1 \cap L_2 \subseteq L(M)$. Comecemos por supor que $w \in \Sigma^*$, arbitrário. Denotemos a sequência de estados gerada por w em M_i por $(r_0^{(i)}, r_1^{(i)}, \ldots, r_n^{(i)})$ para $i \in \{1, 2\}$ e $n \in \mathbb{N}$. Se $w \in L_1 \cap L_2$, pela definição de M_1 e M_2 temos que $r_n^{(1)} \in F_1$ e $r_n^{(2)} \in F_2$. Deste modo concluímos que $\delta(w) = (r_n^{(1)}, r_n^{(2)}) \in F$. Como W é arbitrário, segue que $L_1 \cap L_2 \subseteq L(M)$. Por outro lado, se $w \notin L_1 \cap L_2$, então temos que $r_n^{(1)} \notin F_1$ e $r_n^{(2)} \notin F_2$, o que leva a $(r_n^{(1)}, r_n^{(2)}) \notin F$, portanto $w \notin L(M)$. Logo, $L(M) \subseteq L_1 \cap L_2$, e concluímos que $L(M) = L_1 \cap L_2$.

Exercício 6

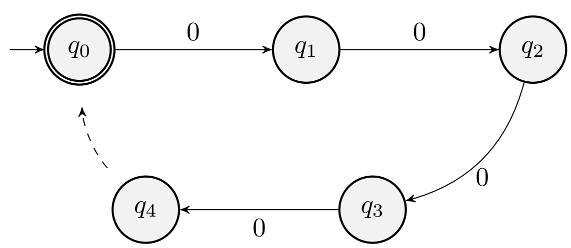
Dada uma string $w = w_1 w_2 \ldots w_n \in \Sigma^*$ definimos o seu reverso $rev(w) = w_n w_{n-1} \ldots w_2 w_1$. Para uma linguagem $L \subseteq \Sigma^*$, definimos $rev(L) = rev(w)|w \in L$. Mostre que se L é regular então $rev(L)$ também é regular.

Resolver com um AFN .-.

Exercício 7

Seja $L_n = \{0^k |\ k$ é múltiplo de $n\}$. Mostre que L_n é regular para qualquer $n \in \mathbb{N}^+$.

Fixamos $n \in \mathbb{N}^+$. Queremos mostrar que o AFD $M = (S, \Sigma, \delta, s, F)$, reconhece L_n *i.e* $L(M) = L_n$.



Vamos definir M. Comecemos pelo conjunto de estados S. Queremos cobrir todos os multiplos possíveis, comecando no q_0, para tal vamos ter $m-1$ estados tal que m é o número de multiplos possíveis ou seja, $S = \{q_0, q_1, q_2, \ldots, q_{m-1}\}$. Intuitivamente $\Sigma = 0$. O estado inicial q será o estado q_0, portanto $s = q_0$. O conjunto de estados finais F é dado por um único estado q_0 o que se traduz para $F = \{q_0\}$. Falta nos então definir δ. Em cada estado $q_i \in S$ queremos ler o símbolo $'0'$ repetindo ciclicamente até chegar ao estado final q_0. Portanto obtemos:
$$\delta(q_i, 0) = q_{(i+1)\%n}$$
Desta forma, M reconhece sequência de potências de 0 cujo expoente seja um múltiplo de n. Portanto agora falta-nos provar que $L(M) = L_n$. Para tal vamos demonstrar que $L(M) \subseteq L_n$ e $L_n \subseteq L(M)$. Comecemos por supor que $w \in \Sigma^*$, arbitrário. Denotemos a sequência de estados gerado por w em M por $(q_0, q_1, q_2, \ldots, q_i)$ com $i \in \mathbb{N}$. Se $w \in L_n$ pela definição de L_n ,$q_i \in F$. Como $F = \{q_0\}$, $q_i = q_0$. Ou seja $\delta(w) = q_0 \in F$. Visto que w é arbitrário, segue que $L_n \subseteq L(M)$. Por outro lado, se $w \notin L_n$ então $q_i \notin F$, pelo que $q_i \neq q_0$. Como $F = \{q_0\}$ e $q_i \neq q_0$, $w \notin L(M)$. Logo, $L(M) \subseteq L_n$, e concluímos que $L(M) = L_n$.