BCC244

Auômato Finito Determinístico

Máquinas de Estados Finitos

- As máquinas de estados finitos são máquinas abstratas que capturam partes essenciais de algumas máquinas concretas.
- Tipos
  - Transdutoras – máquinas com entradas e saída
  - Reconhecedoras – possuem duas saídas possíveis, “aceita” e “rejeita”.
- A memória de uma máquina de estados finitos é limitada e organizada em torno do conceito de “estado”.

Exemplo 1

- Um homem, um leão, um coelho e um repolho devem atravessar um rio usando uma canoa, com a restrição de que o homem deve transportar no máximo um dos três de cada vez de uma margem a outra. Além disso, o leão não pode ficar na mesma margem que o coelho sem a presença do homem, e o coelho não pode ficar com o repolho sem a presença do homem. O problema consiste em determinar se é possível fazer a travessia.

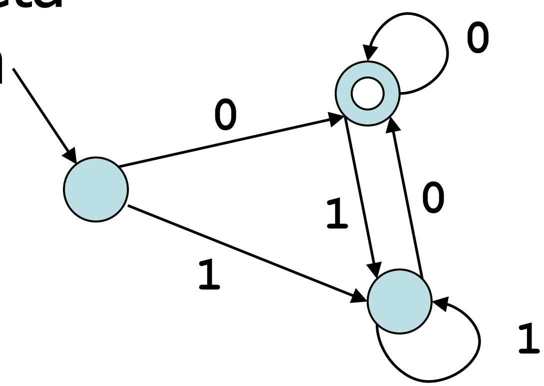
Exemplo 2

- Projetar uma máquina que, dada uma seqüência de 0´s e 1´s, determinar se o número representado por ela na base 2 é divisível por 6.

Autômato Finito Determinístico

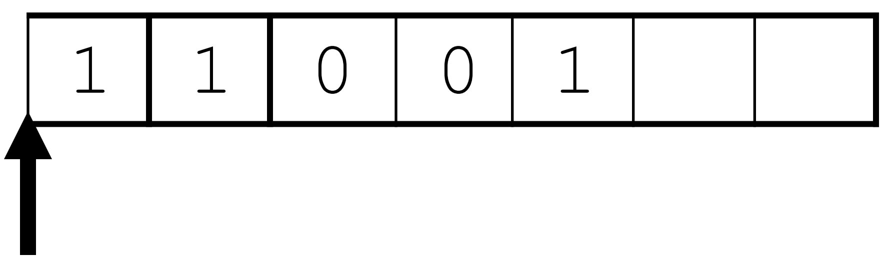
Mais parecido com computer:

esta seta denota inicio

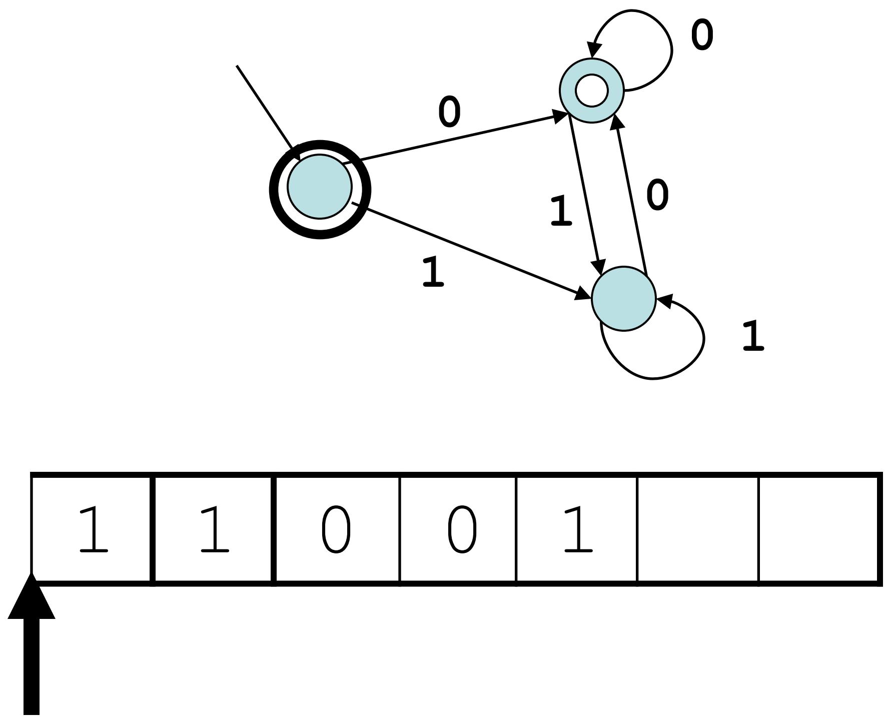


círculo duplo denota aceita

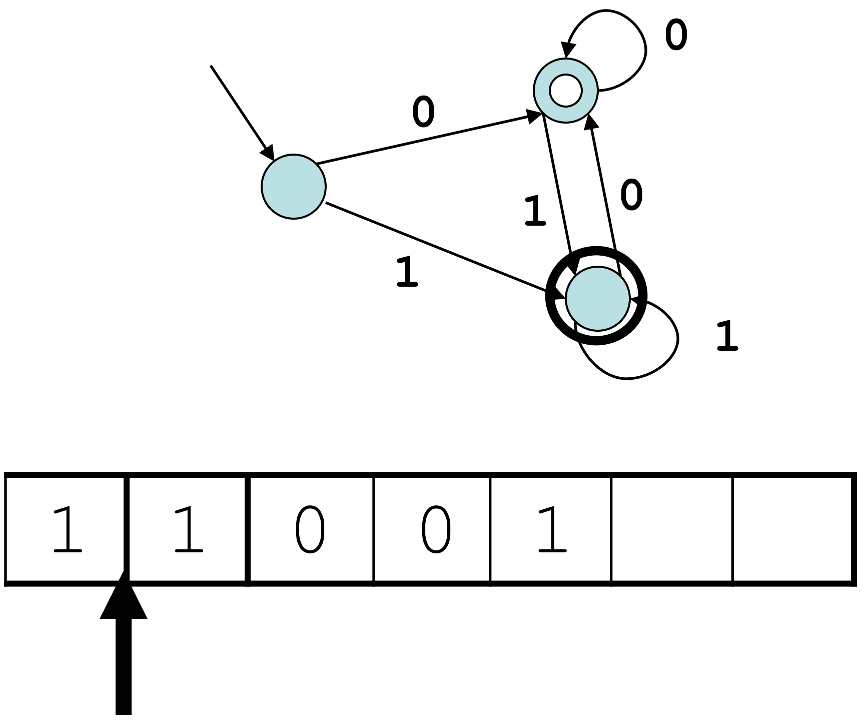
entrada em fita lida da esquerda p/ direita



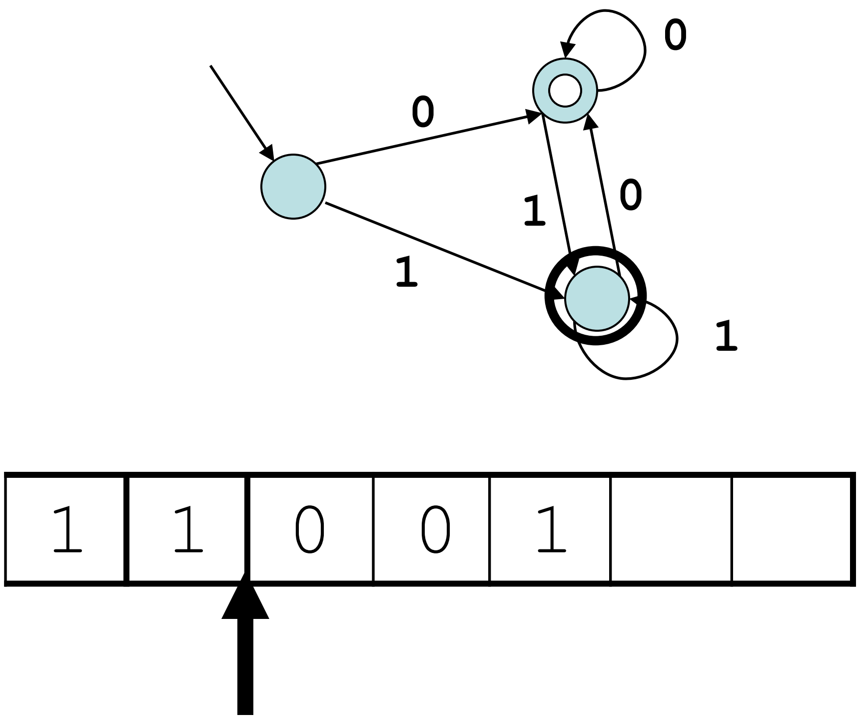
Autômato Finito Determinístico



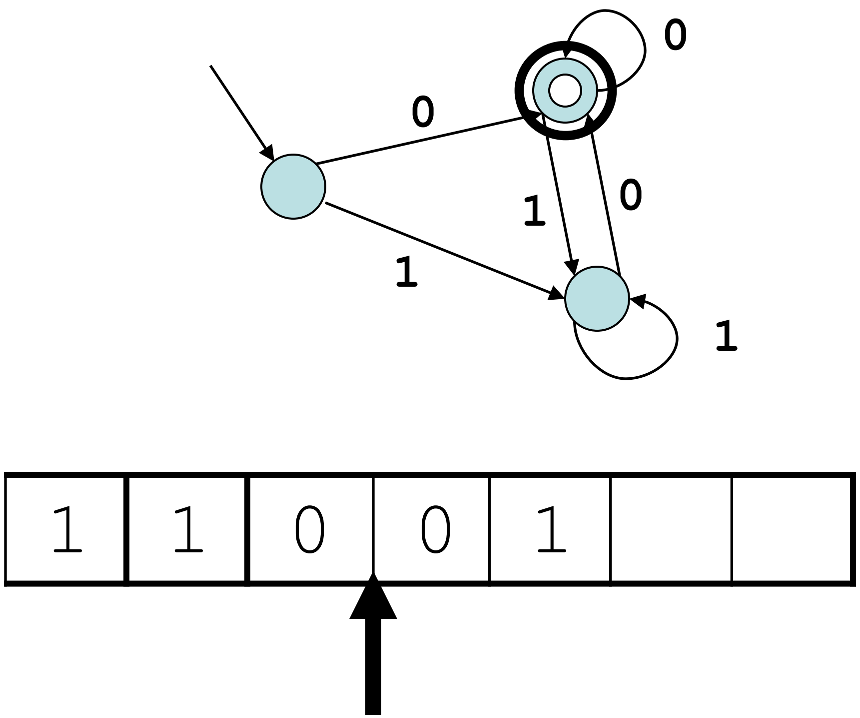
Autômato Finito Determinístico



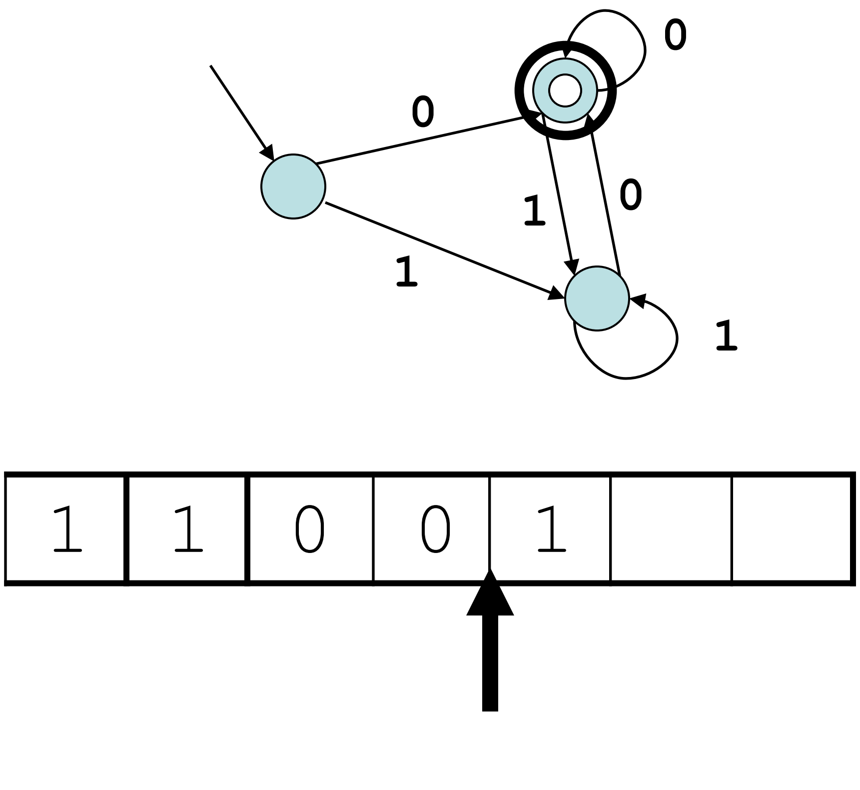
Autômato Finito Determinístico



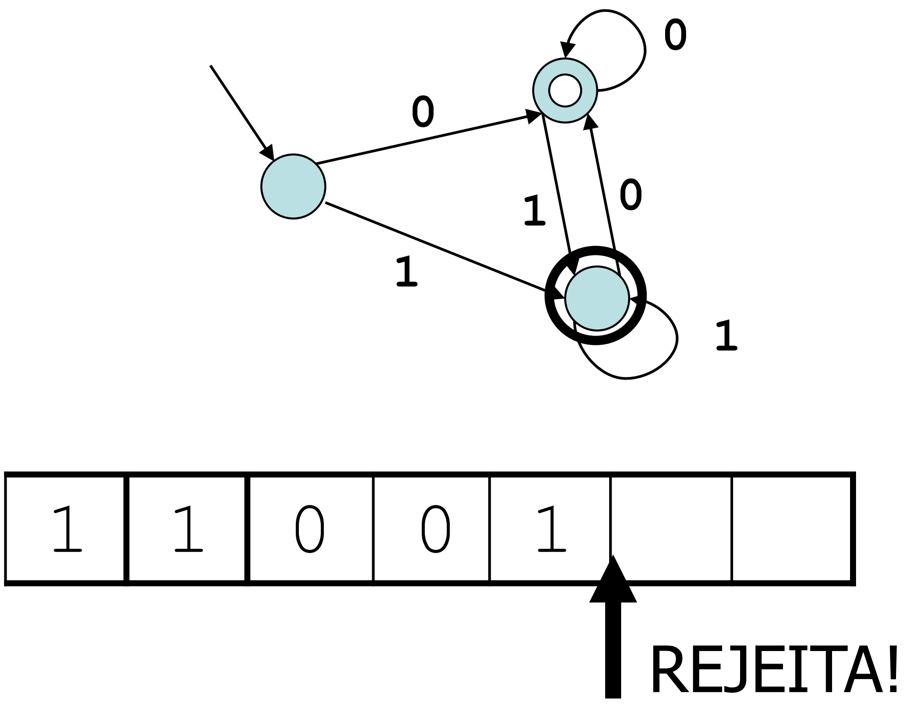
Autômato Finito Determinístico



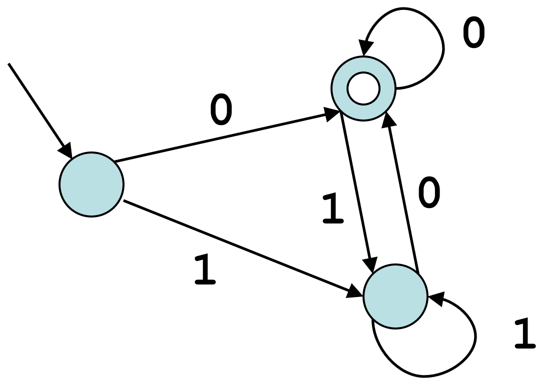
Autômato Finito Determinístico



Autômato Finito Determinístico

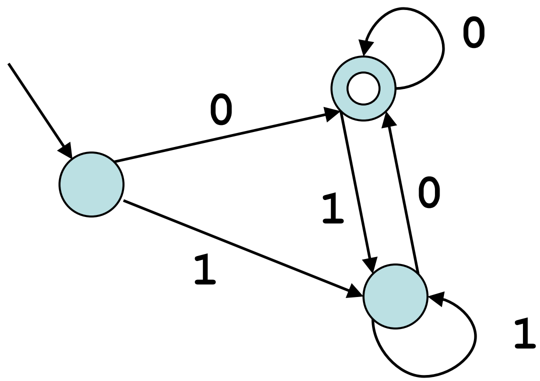


Autômato Finito Determinístico



Q: Que tipos de bitstrings são aceitos?

Autômato Finito Determinístico



R: Bitstrings que representam números binários pares.

Autômato Finito Determinístico

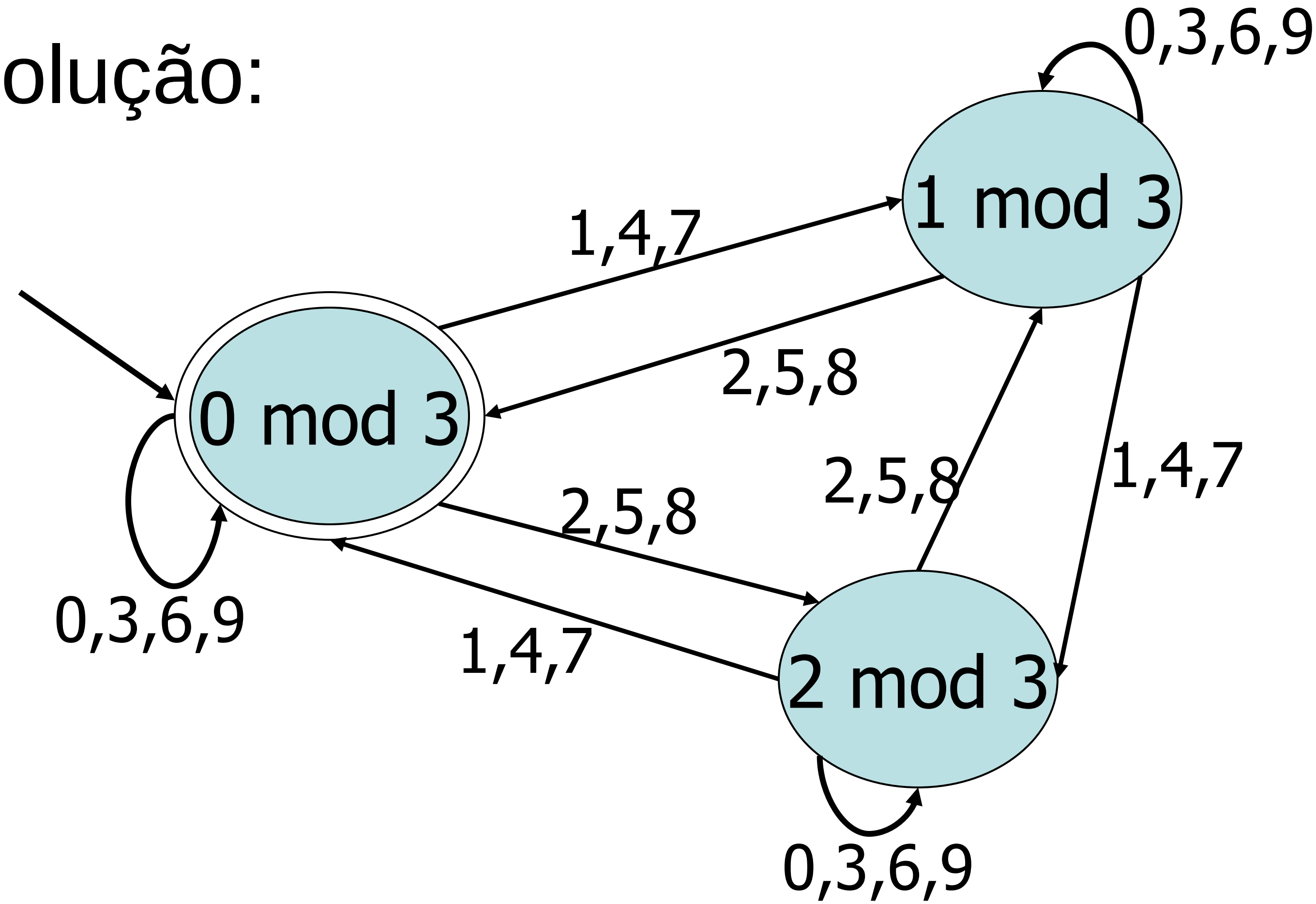
Exercício: Projete uma máquina que determina quando um string de entrada é um número na *base*-10 divisível por 3

Qual deve ser o alfabeto?

Como você pode determinar se um número é divisível por 3?

Autômato Finito Determinístico

Solução:



Definição Formal de FA

DEF: Um ***autômato finito*** *(determinístico)* (**FA**) consiste de um conjunto de **estados** Q, um **alfabeto** Σ, **transições** *rotuladas* entre estados δ, um **estado inicial** $q_0 \in Q$, e um conjunto de **estados de aceitação** F.

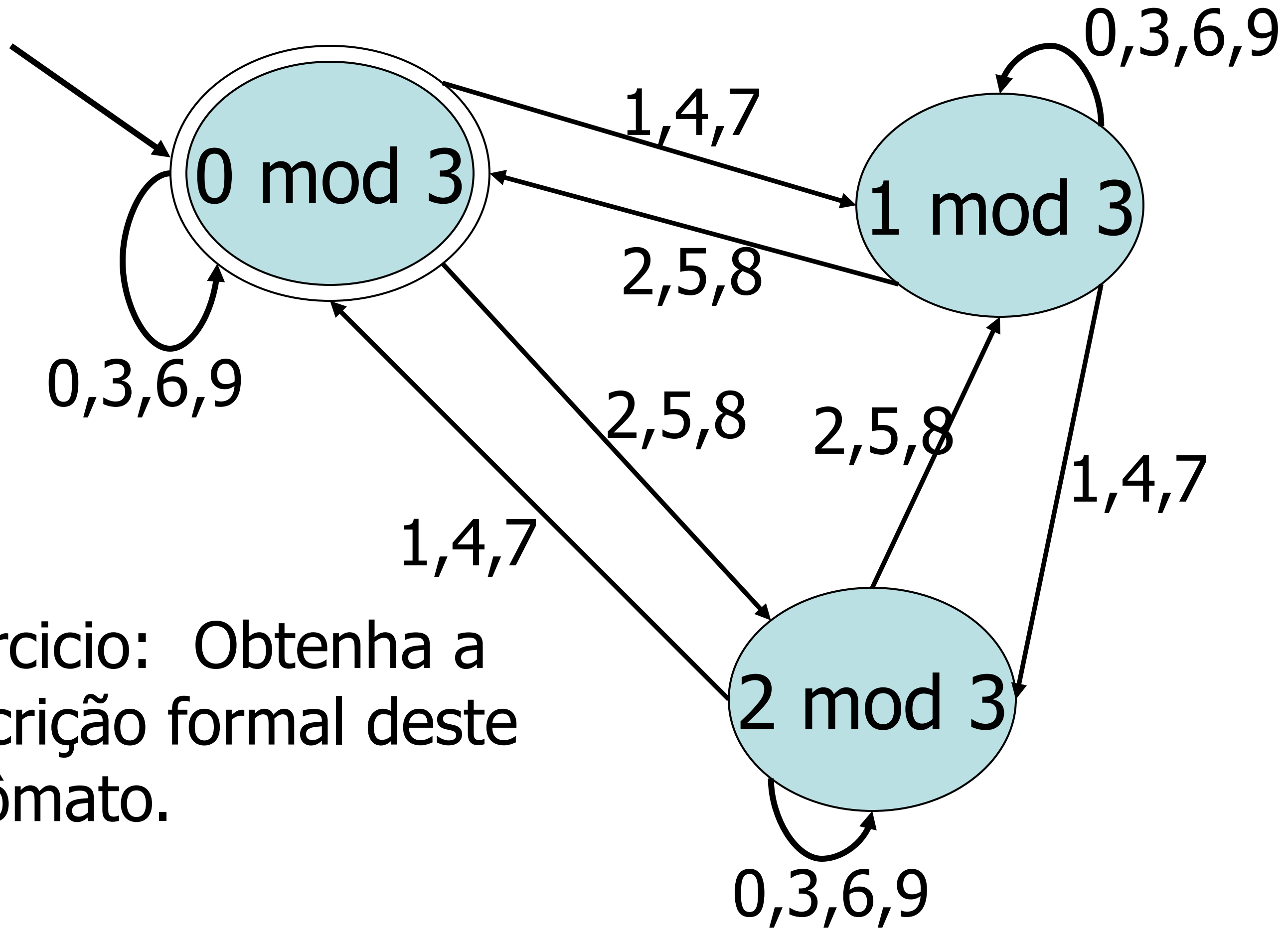
$$M = (Q, \Sigma, \delta, q_0, F)$$

Definição Formal de FA

Note que o string de entrada, assim como a fita que contém o string de entrada, são implícitos na definição de um FA. Ou seja, a definição provê apenas uma visão *estática*. É necessária explicação adicional para entender como um FA interage com a sua entrada.

Porque *Determinístico*?

Determinístico significa que existe informação suficiente para sempre determinar qual é o próximo estado para o qual vai o Autômato, ao ler um dado símbolo. Nosso Exemplo de Máquina de Venda de fato *não* era determinístico porque, depois de terem sido depositados $.45, os efeitos de depósitos adicionais são indefinidos.



Exercicio: Obtenha a descrição formal deste autômato.

Definição de FA, exemplo

Q = { 0 mod 3, 1 mod 3, 2 mod 3 } (renomeie: $\{q_0, q_1, q_2\}$)

Σ = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }

q_0 = 0 mod 3

F = { 0 mod 3 }

δ – requer explicação adicional

A função de transição δ

- δ Determina o estado para o qual vai o autômato, dado o *estado corrente* e o *símbolo corrente* na entrada. I.e., dado um estado $q \in Q$ e um símbolo $a \in \Sigma$, δ define um único *estado alvo* $q' \in Q$. Em outras palavras, δ é uma função do produto Cartesiano $Q \times \Sigma$ em Q :

$$\delta : Q \times \Sigma \to Q$$

A função de transição δ

$$\delta : Q \times \Sigma \rightarrow Q$$

$$\delta(q_0, 2) = q_2,\ \ \delta(q_0, 9) = q_0, \delta(q_1, 2) = q_0,$$

$$\delta(q_1, 7) = q_2,\ \ \delta(q_2, 3) = q_2, \delta(q_2, 5) = q_1.$$

$$\text{Questão} : \delta(q_i, j) = ?$$

A função de transição δ

$$\delta(q_i, j) = q_{(i+j) \bmod 3}$$

Usualmente a função de transição não tem uma definição tal como nesse caso, dada por uma fórmula simples.

Definição Formal de FA: Dinâmica

Como um FA opera sobre um string? Existe implicitamente a noção de uma fita auxiliar que contém o string. O FA lê a fita da esquerda para a direita e cada caractere faz com que o autômato vá para um novo estado, definido pela função δ. Quando o string é lido completamente, ele é aceito ou não, conforme o estado final do FA seja ou não um estado de aceitação.

Definição Formal de FA: Dinâmica

DEF: Um string *u* é ***aceito*** por um autômato sse o caminho a partir do estado inicial q_0 que é rotulado por *u* termina em um estado de aceitação.

Linguagem Aceita por um FA

DEF: A ***linguagem aceita por*** um FA *M* é o conjunto de todos os strings que são aceitos por *M* e é denotada por *L* (*M*).

Intuitivamente, pense em todos os possíveis caminhos que levam do estado inicial a um estado de aceitação do autômato. Então pense em todas as possíveis maneiras de rotular esses caminhos (caso existam múltiplos rótulos em algumas setas).

Linguagens Regulares

Veremos mais adiante que nem toda linguagem pode ser descrita como uma linguagem aceita por um FA. Uma linguagem que é aceita por algum FA exibe um alto grau de regularidade.

DEF: Uma linguagem L é chamada ***linguagem regular*** se existe um FA M tal que

$L = L(M)$.

Função de transição estendida

- Seja um AFD $M=(Q, \Sigma, \delta, q_0, F)$. A função de transição estendida $\hat{\delta}$, é uma função de Q x Σ* para Q, definida recursivamente como:

$$\hat{\delta}(q,\varepsilon) = q$$

$\hat{\delta}(q,ay) = \hat{\delta}(\delta(e,a), y)$ para todo a $\in$ Σ e y $\in$ Σ*.

$$L(M) = \{w \in \sum * \mid \hat{\delta}(q_0, w) \in F\}$$

Algumas propriedades

- Sejam os AFDs $M_1=(Q_1, \Sigma, \delta_1, i_1, F_1)$ e $M_2=(Q_2, \Sigma, \delta_2, i_2, F_2)$. Existem AFDs para as seguintes linguagens.

  (1) $\overline{L(M_1)}$

  (2) $L(M_1) \cap L(M_2)$

  (3) $L(M_1) \cup L(M_2)$

- (1) Pode ser obtido a partir de M_1 simplesmente colocando-se como estados finais aqueles que não são finais em $M_{1.}$

Algumas propriedades

- (2) Seja o AFD $M_3 = (Q_1 x Q_2, \Sigma, \delta_3, [i_1, i_2], F_3)$, construído com
  - $\delta_3([e_1, e_2], a) = [\delta_1(e_1, a), \delta_2(e_2, a)]$
  - $F_3 = F_1 \times F_2$
- (3) Seja o AFD $M_3 = (Q_1 x Q_2, \Sigma, \delta_3, [i_1, i_2], F_3)$, construído com
  - $\delta_3([e_1, e_2], a) = [\delta_1(e_1, a), \delta_2(e_2, a)]$
  - $F_3 = \{[e_1, e_2] \in Q_1 x Q_2 \mid e_1 \in F_1 \text{ ou } e_2 \in F_2\}$

Exercício

- Defina um AFD que aceita a linguagem *L* sobre o alfabeto {0,1} cujos strings possuem tamanho múltiplo de 3 ou terminam com 1.
- Defina um AFD que aceita a linguagem *L* sobre o alfabeto {0,1} cujos strings começam com 0 e terminam com 10 ou com 11