TEORIA DA COMPUTAÇÃO

AULA 3: AUTÔMATOS E LINGUAGENS (CAP 1)

LINGUAGENS REGULARES

PROFESSOR: LUCAS CAMBUIM

Autômatos Finitos (Descrição Formal)

- Diagrama de estado são fáceis de entender intuitivamente.
- Mas, Necessitamos da definição formal por duas razões:
  - Definição formal é precisa
    - Dúvidas como:
      - “autômatos podem ou não ter 0 estados de aceitação?”
      - “autômatos devem ter exatamente uma transição saindo de cada estado para cada símbolo de entrada possível”
    - São sanadas com a definição formal
  - Uma definição formal provê notação.
    - Uma boa notação ajuda a pensar e expressar seus pensamentos claramente.

Autômatos Finitos (Descrição Formal)

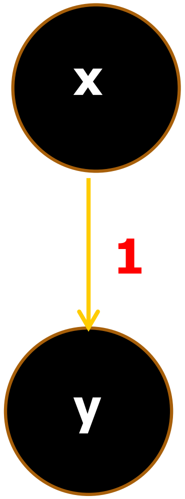
A definição formal define precisamente as partes de um autômato finito:

- conjunto de estados,
- alfabeto de entrada,
- regras para movimentação,
- estado inicial
- e estados de aceitação.

Em linguagem matemática, uma lista de cinco elementos é frequentemente chamada 5-upla.

Função de transição

- Define as regras de movimentação
- Se um autômato finito tem uma seta de um estado x para um estado y rotulada com um símbolo de entrada 1
- Então podemos escrever formalmente com $\delta(x,1) = y$



Definição Formal

Um *autômato finito* é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$, onde

1. Q é um conjunto finito denominado os *estados*,
2. Σ é um conjunto finito denominado *alfabeto*,
3. $\delta: Q \times \Sigma \rightarrow Q$ é a *função de transição,*
4. $q_0 \in Q$ é o *estado inicial*, e
5. $F \subseteq Q$ é o *conjunto de estados de aceitação (ou finais).*

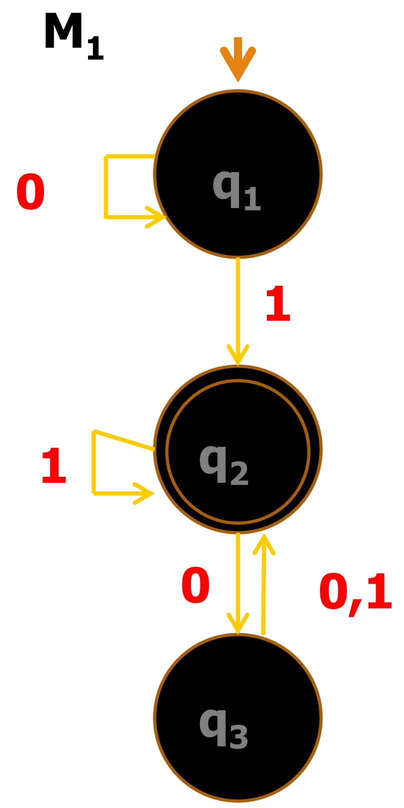
Aplicando as formalidades no diagrama

Podemos escrever M1 formalmente escrevendo M1 = (Q, Σ, δ, q_0, F), onde

1) Q = $\{q_1, q_2, q_3\}$
2) **Σ** = {0,1}
3) δ é descrita como:

	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

4) q0 = q1
5) F = $\{q_2\}$



Autômatos Finitos (Definição formal)

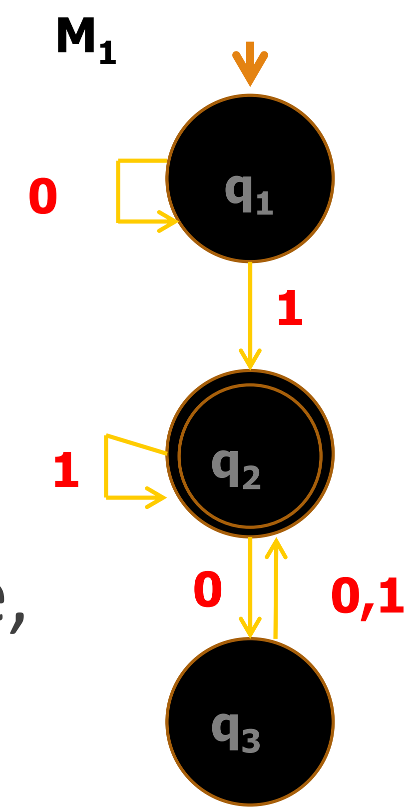
Definição:

Se A é o conjunto de todas as cadeias que a máquina M aceita, dizemos que A é a linguagem da máquina M e escrevemos L(M) = A.

- Dizemos que M *reconhece* A ou que M *aceita* as cadeias de A.
- Uma máquina pode aceitar várias cadeias mas ela sempre reconhece uma única linguagem.
- No exemplo do slide anterior:
- Agora, vamos definir formalmente os AFs dos exemplos anteriores.

Aplicando as formalidades

A = {w | w contém pelo menos 1 e
um número par de 0s segue
o último 1.

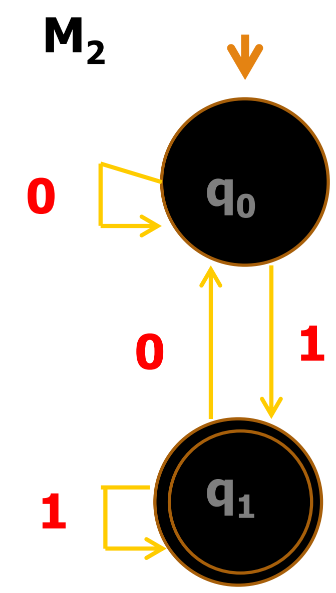


Então L(M_1) = A, ou, equivalentemente,
M_1 reconhece A

Aplicando as formalidades

1) Q = $\{q_0, q_1\}$
2) **Σ** = {0,1}
3) δ é descrita como:

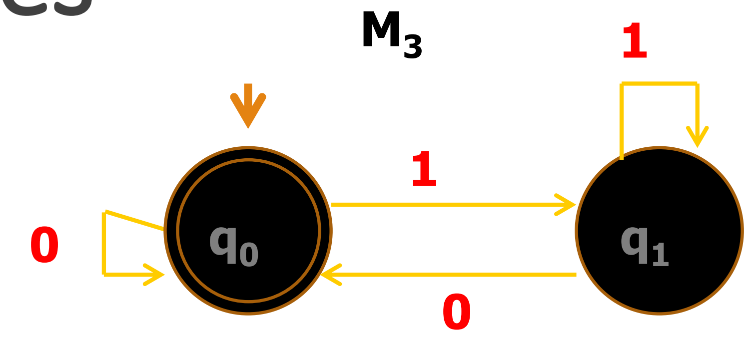
	0	1
q_0	q_0	q_1
q_1	q_0	q_1



4) q0 = q1
5) F = $\{q_1\}$

$L(M_2)$={w | w termina em 1}

Aplicando as formalidades



1) Q = $\{q_0, q_1\}$
2) **Σ** = {0,1}
3) δ é descrita como:

	0	1
q_0	q_0	q_1
q_1	q_0	q_1

4) q0 = q1
5) F = $\{q_0\}$

L(M_3)={w | w é a cadeia vazia ε ou termina em um 0

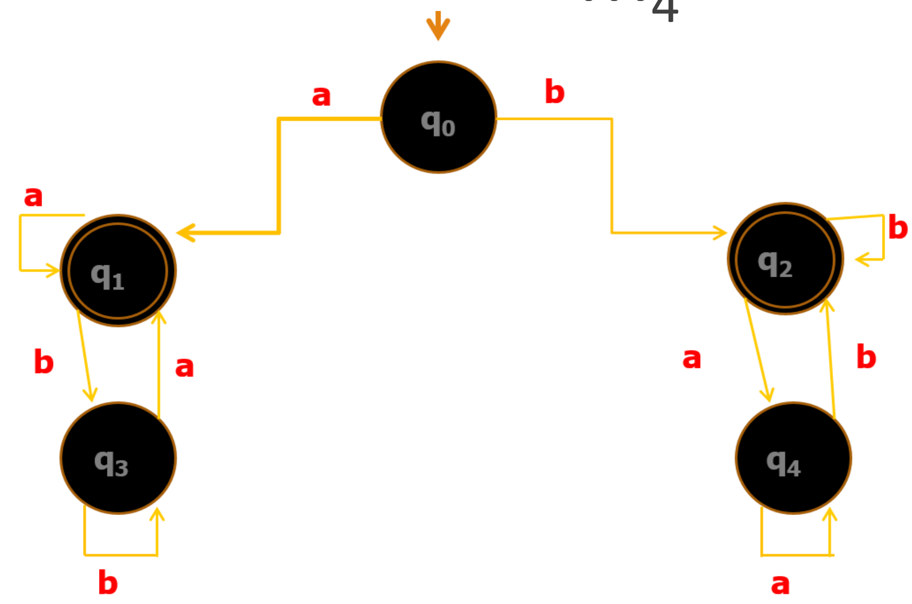
Aplicando as formalidades

1) Q = $\{q_0, q_1, q_2, q_3, q_4\}$
2) **Σ** = {a,b}
3) δ é descrita como:

	a	b
q_0	q_1	q_2
q_1	q_1	q_3
q_2	q_4	q_2
q_3	q_1	q_3
q_4	q_4	q_2

4) $q_0 = q_0$
5) F = $\{q_1, q_2\}$

M_4



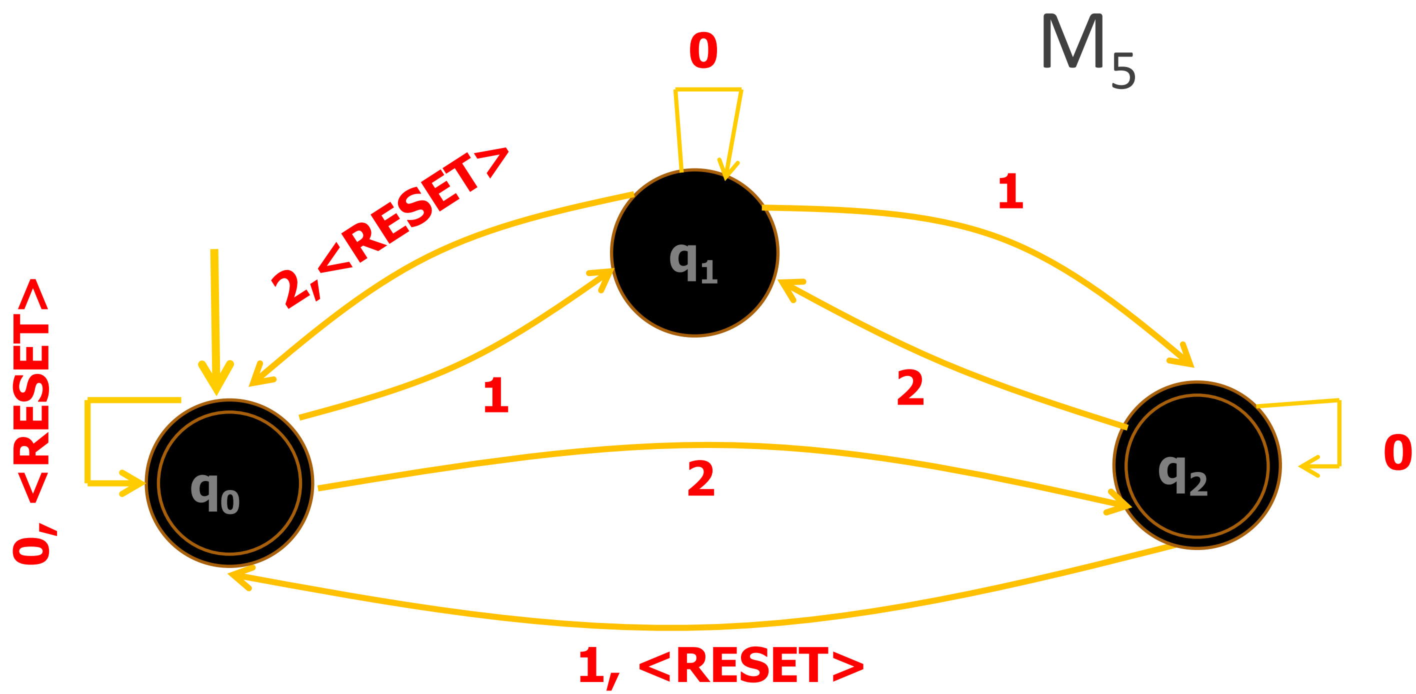
L(M_4)={w | w começa e termina no mesmo símbolo}

Aplicando as formalidades

Exemplo 1.13

Queremos uma máquina com três estados Q = {q0,q1,q2} e o alfabeto **Σ** = {<RESET,0,1,2} que mantenha o contador da soma dos símbolos numéricos de entrada que ele lê. Toda vez que recebe um <RESET> ela reinicia o contador para 0. Ela aceita se a soma for um múltiplo de 3.

Aplicando as formalidades



L(M4)={w | w começa e termina no mesmo símbolo}

Autômatos Finitos (Definição formal)

Situações em que não é possível escrever autômatos finitos por diagrama de estados:

- Quando o diagrama é bem grande
- Quando a descrição formal depende de algum parâmetro não especificado.

Autômatos Finitos (Definição formal)

Exemplo 1.15

Generalização do problema 1.13

Para cada $i \geq 1$ seja A_i, a linguagem de todas as cadeias em que a soma dos números é um múltiplo de i, exceto que a soma é reinicializada para 0 sempre que o símbolo <RESET> aparece.

Autômatos Finitos (Definição formal)

Exemplo 1.15

Descrevemos uma máquina B_i, para reconhecer A_i, formalmente da seguinte forma:

1) $Q_i = \{q_0, q_1, q_2, \dots, q_{\{i-1\}}\}$
2) δ é descrita como:
   - $\delta(q_j, 0) = q_j$
   - $\delta(q_j, 1) = q_k$, onde $k = j + 1$ módulo i
   - $\delta(q_j, 2) = q_k$, onde $k = j + 2$ módulo i
   - $\delta(q_j, \text{<RESET>}) = q_0$

Definição Formal de Computação

Definição formal da computação de um autômato finito

Seja $M = (Q, \Sigma, \delta, q_0, F)$ um autômato finito e suponha que $w = w_1 w_2 \ldots w_n$ seja uma cadeia onde cada w_i é um membro do alfabeto.

M ***aceita*** w se uma sequência de estados $r_0, r_1 \ldots r_n$ em Q existe com três condições:

1. $r_0 = q_0$,
2. $\delta(r_i, w_{i+1}) = r_{i+1}$, para $i = 0, \ldots, n-1$; e
3. $r_n \in F$.

Dizemos que M ***reconhece a linguagem*** A se $A = \{w \mid M \text{ aceita } w\}$.

Definição

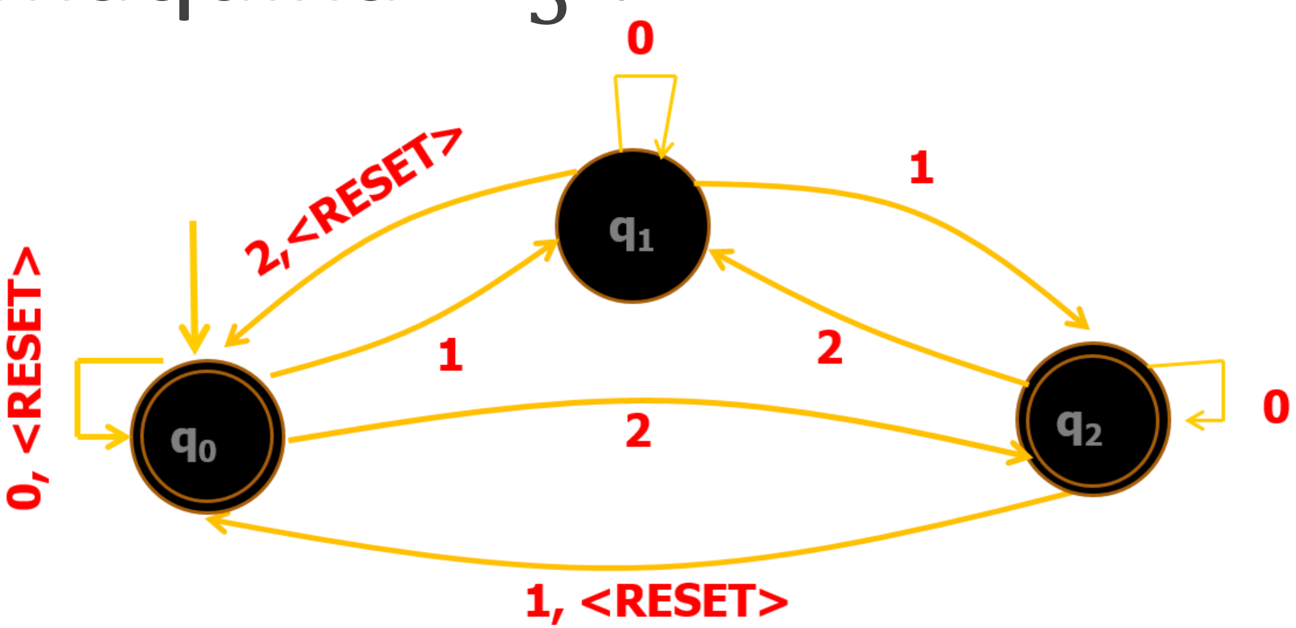
Uma linguagem é chamada de uma ***linguagem regular*** se algum autômato finito a reconhece.

Exemplo 1.17

Seja w a cadeia

10<RESET>22<RESET>012

Seja a máquina M_5 :



Exemplo 1.17

Então M_5 aceita w conforme a definição formal de computação, porque a sequência de estados na qual ela entra quando está computado sobre w é:

$$q_0, q_1, q_1, q_0, q_2, q_1, q_0, q_0, q_1, q_0$$

O que satisfaz as três condições.

Projetando Autômatos Finitos

- Suponha que lhe é dada alguma linguagem e você deseja projetar um AF que a reconheça.
- Faça de conta que você é o autômato. Você recebe uma cadeia de entrada e tem que determinar se ela é um membro da linguagem que o AF é suposto reconhecer.
- Você vai vendo os símbolos na cadeia um por um. Depois de cada símbolo você tem que decidir se a cadeia vista até então está na linguagem. A razão é que você, como a máquina, não sabe quando o final da cadeia está vindo, portanto você tem que estar sempre pronto com a resposta.

Projetando Autômatos Finitos

- De modo a tomar essas decisões, você tem que perceber **o que precisa lembrar** sobre a cadeia na medida em que a está lendo.
- Por que não simplesmente lembrar de tudo que você viu?
- Lembre-se: Autômatos finitos têm um número limitado de estados.
- Felizmente, para muitas linguagens você não precisa lembrar de toda a entrada.

Projetando Autômatos Finitos: Exemplo

- Suponha que alfabeto seja {0,1} e que a linguagem consista de todas as cadeias com um número ímpar de 1s.
- Faça de conta que você é o autômato E_1, que reconhece essa linguagem. Você recebe uma cadeia de entrada de 0s e 1s.
- Você precisa lembrar a cadeia inteira vista até então para determinar se o número de 1s é ímpar?

Projetando Autômatos Finitos: Exemplo

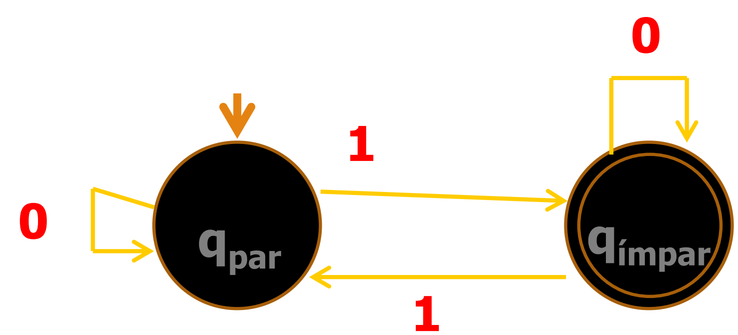
Basta lembrar se o número de 1s visto até então é par ou ímpar e manter essa informação à medida que lê novos símbolos.

Você representa essa informação como uma lista finita de possibilidades:

1. par até agora, e
2. ímpar até agora.

Projetando Autômatos Finitos: Exemplo

- Aí então você atribui um estado a cada uma das possibilidades
- Em seguida, você atribui as transições vendo como ir de uma possibilidade para a outra ao ler um símbolo.
- Em seguida, você coloca como sendo estado inicial aquele correspondendo à possibilidade associada como ter visto 0 símbolos até então.
- Por último, ponha como estados de aceitação aqueles correspondendo a possibilidade nas quais você deseja aceitar a cadeia de entrada.



Projetando Autômatos Finitos: Outro exemplo

Para reconhecer a linguagem de todas as cadeias binárias que contem 001 como uma subcadeia.

Como você reconheceria essa linguagem estivesse fazendo de conta ser E_2?

- Você inicialmente saltaria sobre todos os 1s.
- Se você chegar num 0, então você pode ter acabado de ver o primeiro dos três símbolos no padrão 001
- Se nesse ponto você vê um 1, houve muito poucos 0s, portanto você volta a saltar sobre 1s. Mas, se você vê um 0 nesse ponto, você deve lembrar que você acabou de ver dois símbolos do padrão.
- Agora você precisa encontrar um 1.
- Se você o encontrar, logo você conseguiu achar o padrão. Continue lendo a cadeia de entrada até que você chegue no final.

Projetando Autômatos Finitos: Outro exemplo

Portanto, existem 4 possibilidades:

1. não tem visto quaisquer símbolos do padrão,
2. acaba de ver um 0,
3. acaba de ver 00, ou
4. acaba de ver o padrão inteiro 001.