

IBM1088 Linguagens Formais e Teoria da Computação

Linguagens Regulares

Evandro Eduardo Seron Ruiz
evandro@usp.br

UNIVERSIDADE DE SÃO PAULO

A nossa maior glória não reside no fato de nunca cairmos, mas sim em levantarmo-nos sempre depois de cada queda.

Confúcio^a

^aFilósofo chinês (551 a.C. – 479 a.C.).

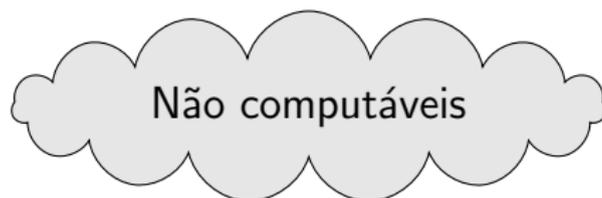
Recordando os objetivos da disciplina

- Principal: Entender a computação
- Criar um **modelo computacional**
- Modelos de linguagem e de máquinas
- Usar **Máquina de estados finitos** ou **Autômato finito**
 - Simplificação de um computador
 - Memória bastante reduzida

Conteúdo

- 1 Introdução
- 2 Sistemas de estados finitos
- 3 Autômatos finitos
 - Introdução
 - Exemplo de autômato finito
 - Formalização
 - Exemplos

Hierarquia de Chomsky



Recursivamente enumeráveis

Sensíveis ao contexto

Livres do contexto

Regulares

Gramáticas

Máquina de Turing

Autômato linearmente limitado

Autômato com pilha

Autômato de estado finito

Máquinas

Linguagens regulares (Tipo 3)

Podemos, e iremos, analisar este tópico de 3 maneiras distintas:

Autômato finito

Expressão regular

Gramática regular

Linguagens regulares (Tipo 3)

Podemos, e iremos, analisar este tópico de 3 maneiras distintas:

Autômato finito

Expressão regular

Gramática regular

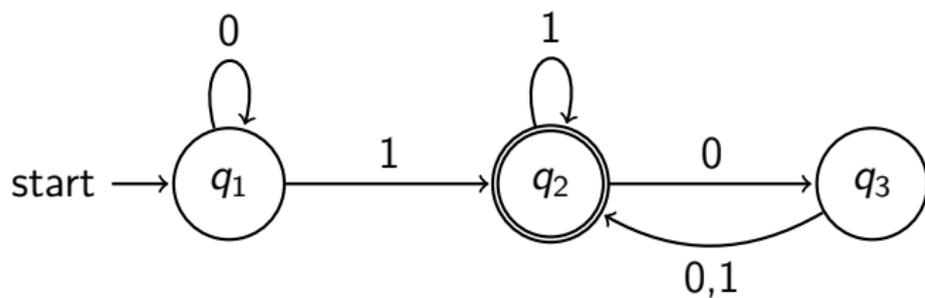
Ou seja, podemos estudar as LR através de autômatos finitos, de expressões regulares ou de gramáticas regulares.

Linguagens regulares

Autômato finito

- Sistema de estados finitos, Máquina de estados finitos
- Formalização de um reconhecedor (Análise Léxica)

Exemplo de um autômato finito



Linguagens regulares

Expressão regular

- Formalismo de notação (Geração da expressão)
- Reconhecedor de expressões (Identificação da expressão)
- Exemplos: \$ ls *.pdf, psicodélic(a|o)

Linguagens regulares

Gramática regular

- Formalismo axiomático
- Gerador de gramática
- Gramática com restrições nas regras de produção

+ sobre Linguagens regulares

- Linguagens mais simples na hierarquia de Chomsky
- Algoritmos de geração e reconhecimento são simples

+ sobre Linguagens regulares

- Linguagens mais simples na hierarquia de Chomsky
- Algoritmos de geração e reconhecimento são simples

Limitações

Por exemplo, não reconhece duplo balanceamento, ou seja, expressões tipo $a^n b^n$.

Vantagens e utilidades do estudo de LR

- Construção e análise de novas linguagens regulares ou reconhecedores
- Algoritmos relativos à LR: simples e eficientes
- Apoio a sistemas:
 - Compiladores: identificação de *tokens*
 - Editores: achar e substituir
 - Animação
 - Hipertexto
 - Hipermissão

Sistemas de estados finitos

Ou *máquinas de estados finitos*, ou autômatos

- Modelo matemático com entradas e saídas definidas
- Definido por **estados** e **ações**
- Número finito e pré-definido de estados

Sistemas de estados finitos

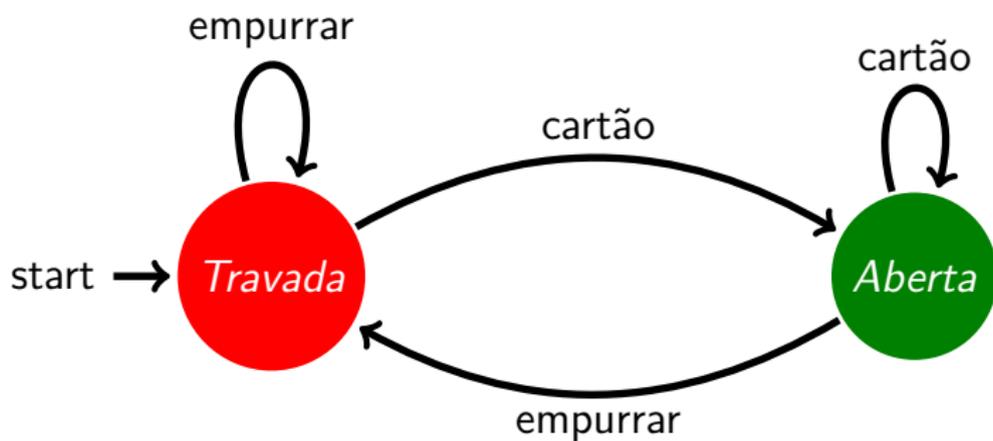
Ou *máquinas de estados finitos*, ou autômatos

- Modelo matemático com entradas e saídas definidas
- Definido por **estados** e **ações**
- Número finito e pré-definido de estados
- **Estado**: mostra a situação atual e passada do sistema
- **Ação**: determina a transição de um estado para outro

Exemplos de autômatos

- Elevador
- Máquina automática de venda
- Semáforos
- Alguns eletrodomésticos simples, e.g.: lavadora de roupas

Exemplo: catraca liberada por cartão (ônibus)



Composição de sistemas

Modos de construção de sistemas a partir de sistemas conhecidos

Seqüencial A execução de um componente depende do término do anterior

Composição de sistemas

Modos de construção de sistemas a partir de sistemas conhecidos

Seqüencial A execução de um componente depende do término do anterior

Concorrente Componentes do sistema trabalham de forma independente. Ordem de execução não é importante

Não-determinista Escolha do próximo componente pode ser variável

Interna Sistema escolhe aleatoriamente

Externa Outro sistema escolhe

Exemplos: Composição de sistemas

Seqüencial Fila de atendimento

Exemplos: Composição de sistemas

Seqüencial Fila de atendimento

Concorrente Atendimento de caixas num banco.
Um atendimento não depende do outro.

Não-determinista Dois ou mais atendentes livres.
O próximo a ser atendido escolhe o atendente.

Composição nas linguagens formais

- Modelos **seqüencial** e **não-determinista** são os mais importantes
- Não-determinismo como capacidade de reconhecer linguagens

Composição nas linguagens formais

- Modelos **seqüencial** e **não-determinista** são os mais importantes
- Não-determinismo como capacidade de reconhecer linguagens

Veremos primeiramente os **autômatos finitos**:

Autômatos finitos

- Sistemas com número **finito** e pré-determinado de estados
- Modelo computacional útil e comum a muitas disciplinas
 - Compiladores
 - LFA
 - Processamento de linguagem
 - Automação, entre outras

Vamos a um exemplo. . .

Ilustração de porta automática

Esta porta automática que permite somente a entrada de pessoas num ambiente, tem um controlador que possibilita sua abertura assim que alguém pisa no tapete frontal

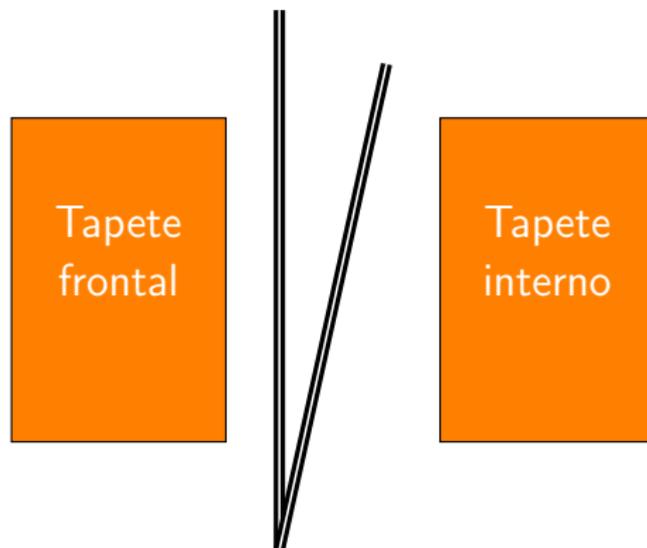


Figura : Ilustração de uma porta automática.

Explicação:

Este tipo de porta já foi muito comum em ambientes comerciais. Normalmente operam em pares, uma porta para entrada e outra para a saída de clientes. A porta se abre ao pisar no tapete frontal e se mantém aberta ao pisar no tapete interno.

Sinais de porta automática

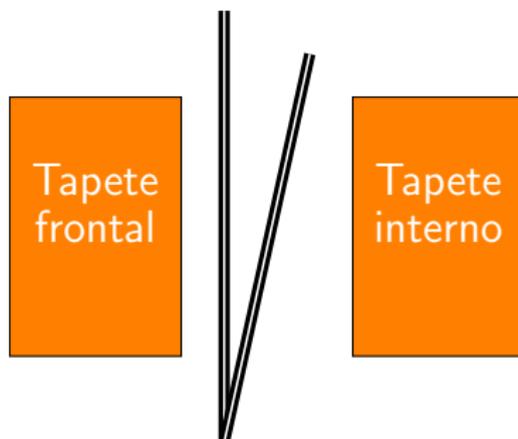
Existem 4 condições possíveis

Frontal Alguém pisa no tapete frontal

Interno Alguém pisa no tapete interno

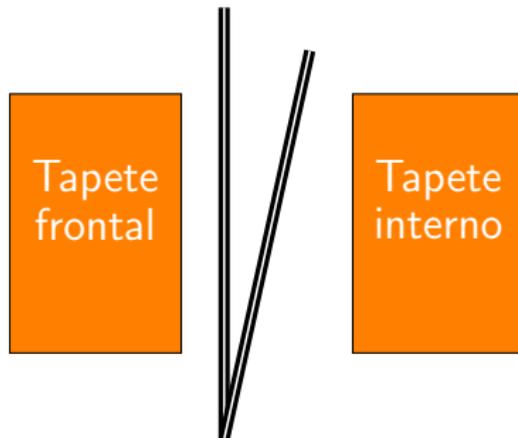
Nenhum Ninguém pisa sobre qualquer tapete

Ambos Pessoas pisam sobre os dois tapetes



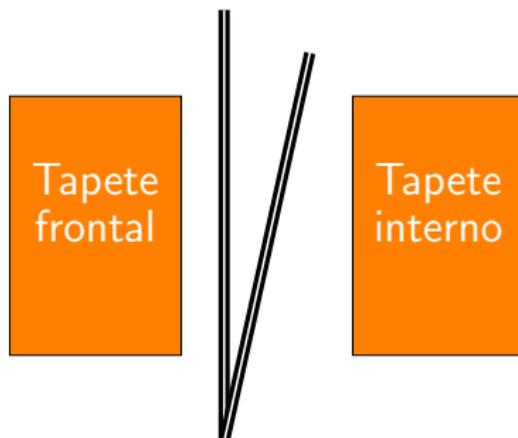
Como o controlador opera? Preencha os espaços com 'aberta' ou 'fechada'

Estado	Sinal de entrada			
	Nenhum	Frontal	Interno	Ambos
Fechada				
Aberta				



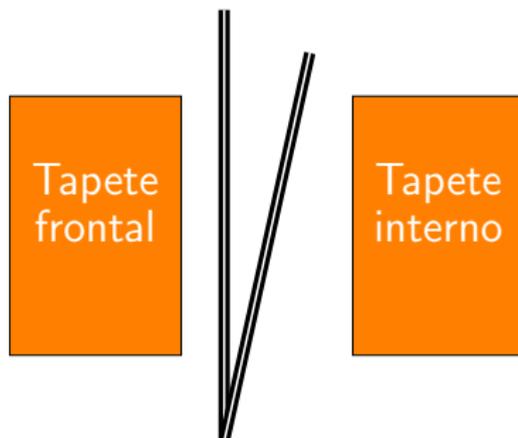
Como o controlador opera? Por exemplo. . .

Estado	Sinal de entrada			
	Nenhum	Frontal	Interno	Ambos
Fechada	fechada	aberta		
Aberta				



Como opera o controlador

Estado	Sinal de entrada			
	Nenhum	Frontal	Interno	Ambos
Fechada	fechada	aberta	fechada	fechada
Aberta	fechada	aberta	aberta	aberta



Porta automática: diagrama de estados



Figura : Considerando só os estados possíveis de uma porta automática.

Porta automática: diagrama de estados

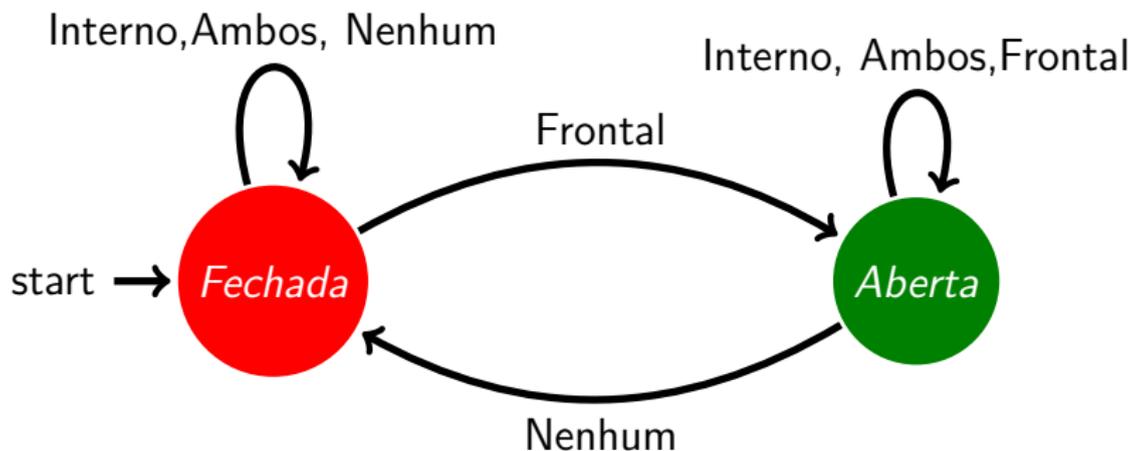


Figura : Diagrama de estados para um controlador de uma porta automática.

Porta automática: diagrama de estados

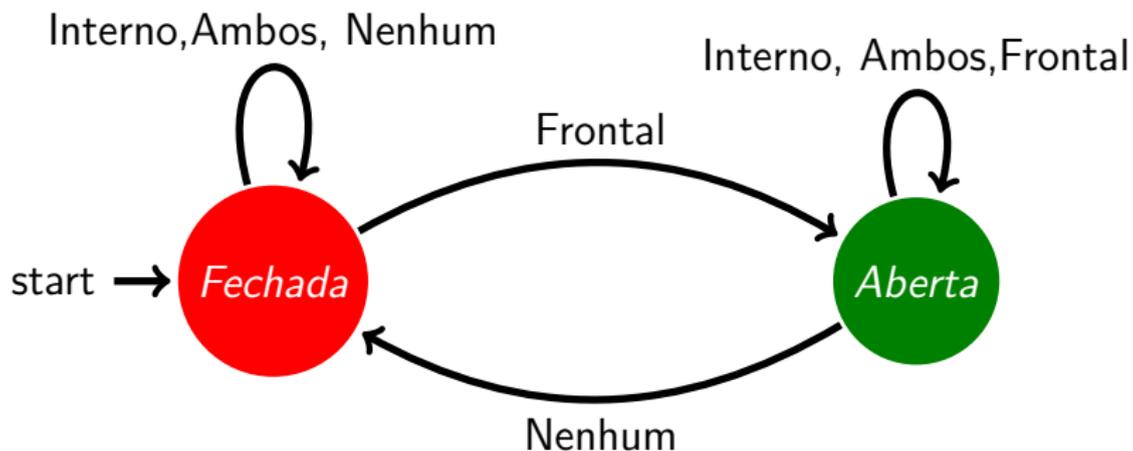
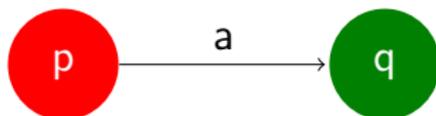


Figura : Diagrama de estados para um controlador de uma porta automática.

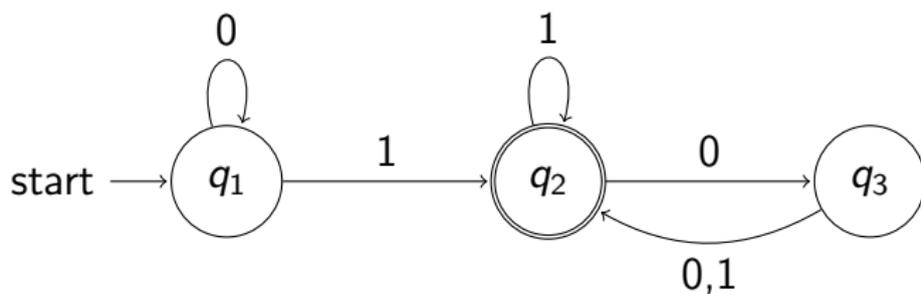
Este é um **autômato finito** de dois estados.

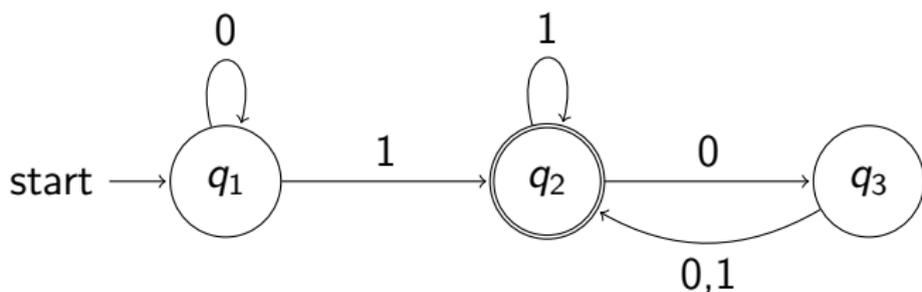
Entendendo o diagrama



Detalhes:

- **p** estado anterior
- **a** símbolo lido
- **q** estado novo
- Função de transição: $\delta(p, a) = q$

Exemplo de um autômato finito M_1 Figura : Um autômato finito de 3 estados, M_1 .

Exemplo de um autômato finito M_1 Figura : Um autômato finito de 3 estados, M_1 .

Características do **diagrama de estado** de M_1 :

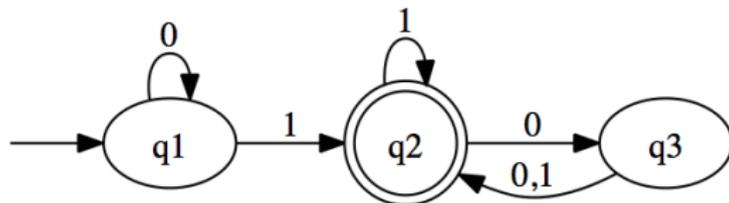
- 3 estados, q_1, q_2, q_3
- **Estado inicial** = q_1
- **Estado de aceitação** = q_2
- **Transições**, setas de um estado para outro

Autômato finito em ação

M_1 aceita ou rejeita entrada.

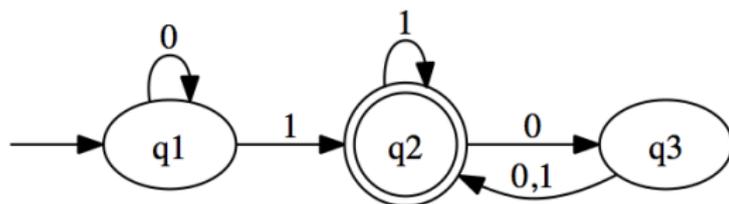
Entrada: 1 1 0 1

- Inicia em q_1
- Le 1, transição de q_1 para q_2
- Le 1, transita q_2 para q_2
- Le 0, transição de q_2 para q_3
- Le 1, transição de q_3 para q_2
- Aceita pois M_1 está no estado de aceitação q_2



Notem que...

- Existem duas possibilidades de transição nos estados
- Estas possibilidades dão origem a uma árvore de opções
- Estas árvores também são características de autômatos não-determinísticos
- Veremos este tópico adiante

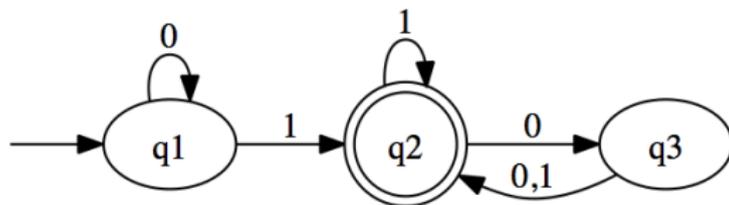


Autômato finito e linguagens

Linguagens aceitas por M_1

- Cadeias como: 1, 11, 101, 100, 0100, 01010000
- Qualquer cadeia que termine com 1
- Cadeias que terminem com número par de zeros após o último 1

Rejeita: 0, 10, 101000...

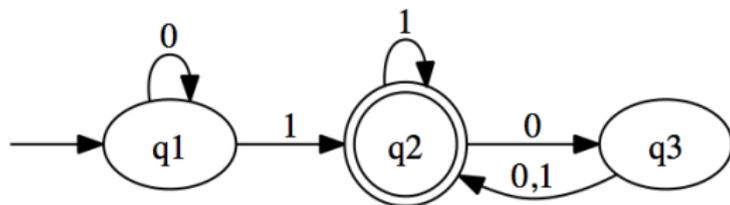


Autômato finito e linguagens

Linguagens aceitas por M_1

- Cadeias como: 1, 11, 101, 100, 0100, 01010000
- Qualquer cadeia que termine com 1
- Cadeias que terminem com número par de zeros após o último 1

Rejeita: 0, 10, 101000...



Pergunta: Podemos descrever uma linguagem que M_1 aceita?

Definição formal

Definição

Um autômato finito determinístico (**AFD**) é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$, em que:

- 1 Q é um conjunto finito de **estados**;
- 2 Σ é um conjunto finito chamado **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**.

Exemplo

Formalizando o autômato $M_1 = (Q, \Sigma, \delta, q_0, F)$ temos:

1 $Q = \{q_1, q_2, q_3\};$

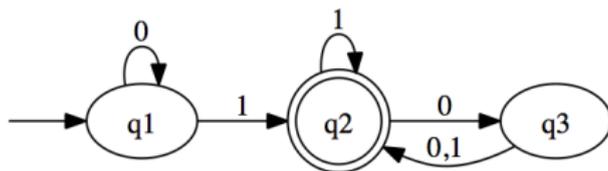
2 $\Sigma = \{0, 1\};$

3 $\delta : Q \times \Sigma \rightarrow Q$ é descrita como:

	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

4 q_1 é o estado inicial; e

5 $F = \{q_2\}.$



Importante!

- Seja A o conjunto de todas as cadeias (palavras) aceitas por M_1 , dizemos que A é a **linguagem de máquina** M_1 e escrevemos $L(M_1) = A$;

Importante!

- Seja A o conjunto de todas as cadeias (palavras) aceitas por M_1 , dizemos que A é a **linguagem de máquina** M_1 e escrevemos $L(M_1) = A$;
- Dizemos que M_1 **reconhece** ou *aceita* A ;
- Uma máquina M reconhece uma única linguagem. Inclusive pode ser unicamente a linguagem vazia ϕ ;

Importante!

- Seja A o conjunto de todas as cadeias (palavras) aceitas por M_1 , dizemos que A é a **linguagem de máquina** M_1 e escrevemos $L(M_1) = A$;
- Dizemos que M_1 **reconhece** ou *aceita* A ;
- Uma máquina M reconhece uma única linguagem. Inclusive pode ser unicamente a linguagem vazia ϕ ;
- Os autômatos também podem ser usados para **gerar cadeias**, **gerar palavras** numa linguagem.

Linguagem de M_1

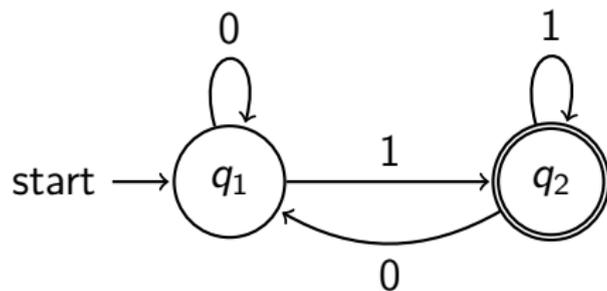
No exemplo anterior, $L(M_1) = A$, sendo

$$A = \{w \mid w \text{ contém pelo menos um } 1 \text{ e} \\ \text{um número par de } 0\text{s} + \text{ último } 1\}$$

ou, podemos dizer que M_1 reconhece A

Exemplo 2

Descrevam o autômato M_2 abaixo como uma 5-upla $(Q, \Sigma, \delta, q_0, F)$.



recordando... Definição formal

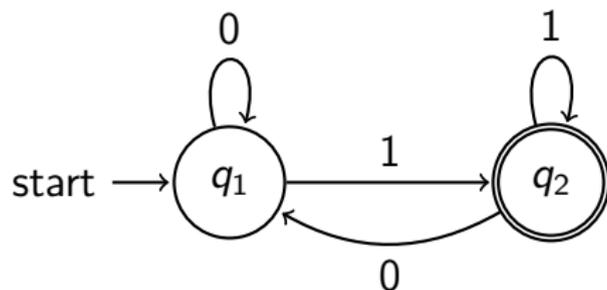
Definição

Um autômato finito determinístico (**AFD**) é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$, em que

- 1 Q é um conjunto finito de **estados**;
- 2 Σ é um conjunto finito chamado **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**.

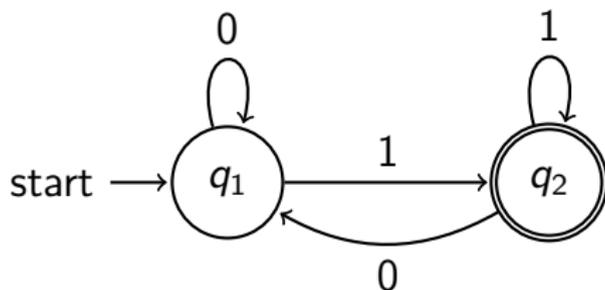
Exemplo 2: Agora façam a descrição

Descrevam o autômato M_2 abaixo como uma 5-upla $(Q, \Sigma, \delta, q_0, F)$.



Exemplo 2: Agora façam a descrição

Descrevam o autômato M_2 abaixo como uma 5-upla $(Q, \Sigma, \delta, q_0, F)$.



1 $Q = \{q_1, q_2\};$

2 $\Sigma = \{0, 1\};$

3 $\delta : Q \times \Sigma \rightarrow Q$ é descrita como:

	0	1
q1	q1	q2
q2	q1	q2

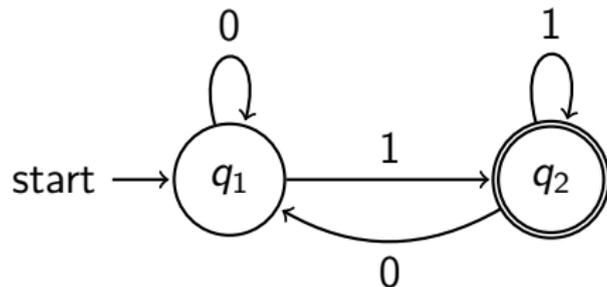
4 q_1 é o estado inicial; e

5 $F = \{q_2\}.$

Testando o exemplo 2

Teste o autômato M_2 abaixo para as seguintes cadeias:

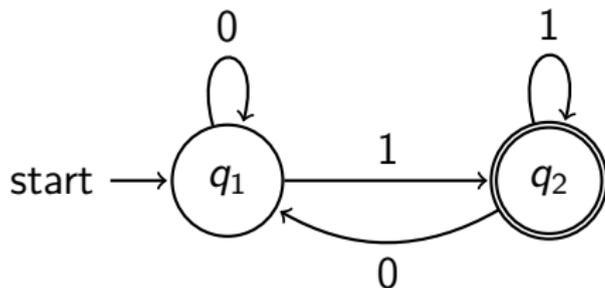
- $c1 = 1\ 1\ 0\ 1$
- $c2 = 1\ 1\ 0$



Testando o exemplo 2

Teste o autômato M_2 abaixo para as seguintes cadeias:

- $c1 = 1\ 1\ 0\ 1$
- $c2 = 1\ 1\ 0$



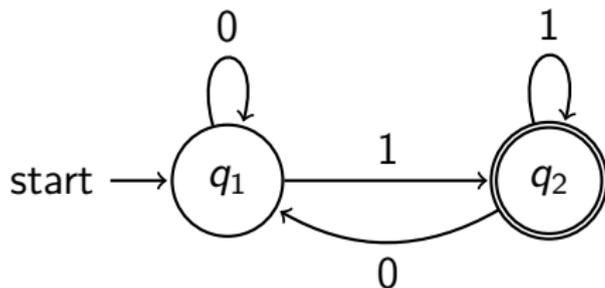
Resposta: Aceita $c1$ e rejeita $c2$.

- 1 Qual é a linguagem de máquina L de M_2

Testando o exemplo 2

Teste o autômato M_2 abaixo para as seguintes cadeias:

- $c1 = 1\ 1\ 0\ 1$
- $c2 = 1\ 1\ 0$

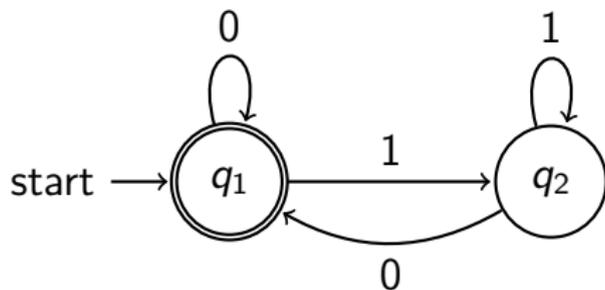


Resposta: Aceita $c1$ e rejeita $c2$.

- 1 Qual é a linguagem de máquina L de M_2
- 2 $L(M_2) = \{w \mid w \text{ termina com } 1\}$

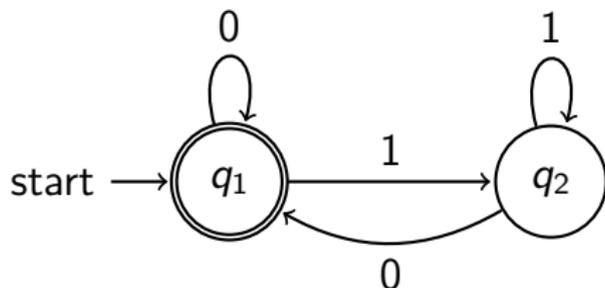
Exemplo 3: exercício

Qual é a linguagem L que o autômato M_3 abaixo reconhece?
Repare que o estado inicial q_1 é um estado de aceitação.



Exemplo 3: exercício

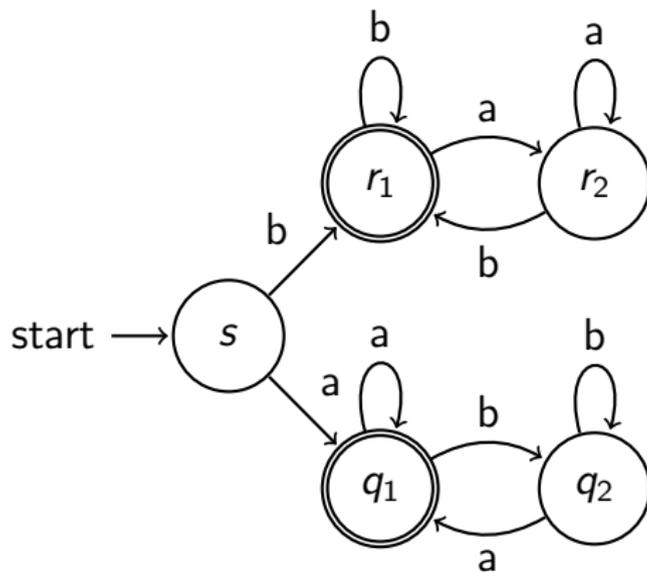
Qual é a linguagem L que o autômato M_3 abaixo reconhece?
Repare que o estado inicial q_1 é um estado de aceitação.



$$L(M_3) = \{w \mid w \text{ é cadeia vazia } \epsilon \text{ ou termina em } 0\}$$

Exemplo 4

Agora temos uma máquina M_4 de cinco estados



Exemplo M_4

- Percebam que ela aceita as cadeias a , b , aa , bb , bab
- Não aceita as cadeias ab , ba , $bbba$

Exemplo M_4

- Percebam que ela aceita as cadeias a , b , aa , bb , bab
- Não aceita as cadeias ab , ba , $bbba$
- Esta máquina nunca retorna ao estado inicial s

Exemplo M_4

- Percebam que ela aceita as cadeias a , b , aa , bb , bab
- Não aceita as cadeias ab , ba , $bbba$
- Esta máquina nunca retorna ao estado inicial s
- Se símbolo inicial for 'a' percorre o caminho inferior e aceita cadeia que termina com 'a'
- Se for 'b', percorre o caminho superior. Aceita cadeia que termina com 'b'
- Portanto, aceita cadeias que termina com o mesmo símbolo de entrada (inicial)

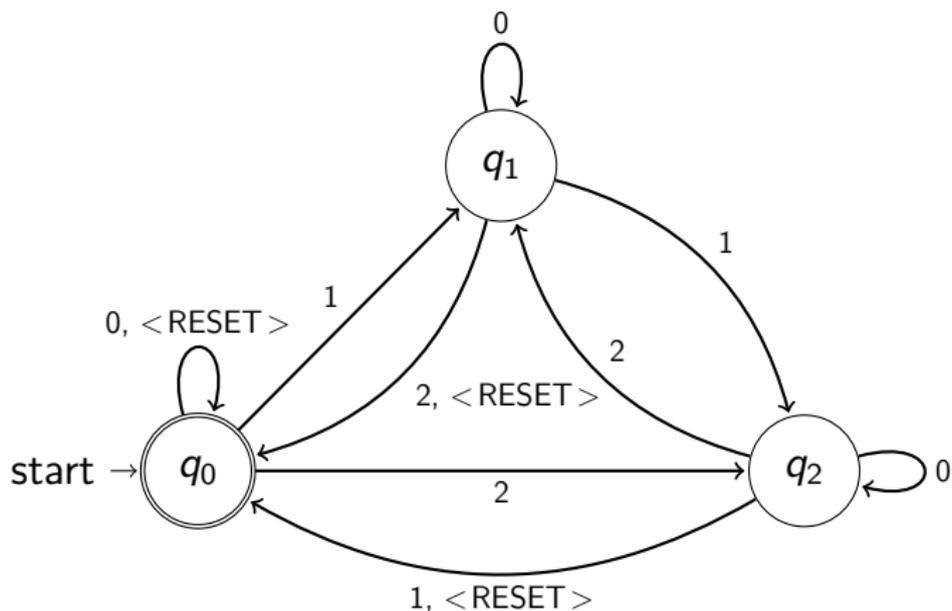
Exemplo 5

Agora temos uma máquina M_5 com um alfabeto de entrada de 4 símbolos

$$\Sigma = \{ \langle \text{RESET} \rangle, 0, 1, 2 \}$$

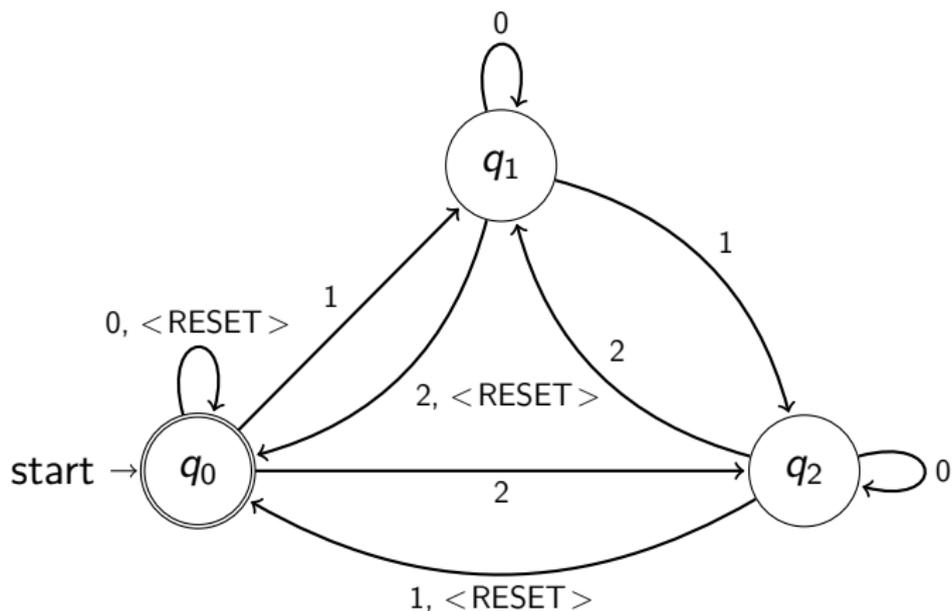
Notem que $\langle \text{RESET} \rangle$ é um símbolo único.

Exemplo 5



Que tipos de cadeia de entrada M_5 aceita?

Exemplo 5



R: M_5 aceita cadeias cuja **soma dos dígitos é múltipla de 3**

Caros,

Continuaremos neste assunto no próximo encontro!