Teoria de Linguagem

Autômatos Finitos Determinísticos

Vinicius H. S. Durelli
✉ durelli@ufsj.edu.br

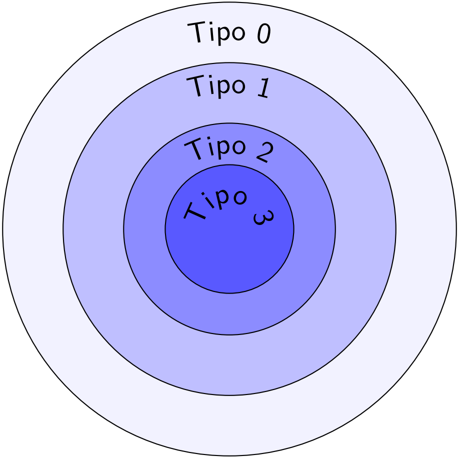
UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL-REI

Organização

Linguagens regulares (Tipo 3)

De acordo com a hierarquia de Chomsky (Chomsky 1956), as linguagens regulares são a **classe de linguagem mais simples**.

- Fortes limitações de expressividade.
- Por exemplo, linguagens que possuem duplo balanceamento não são regulares.



Formalismos usados para “processar” linguagens regulares

Essencialmente, dois formalismos são usados:

- Denotacional; e
- Operacional.

Definição → *Formalismo Operacional*

Máquina abstrata baseada em estados, em instruções primitivas e na especificação de como cada instrução modifica cada estado (Menezes 2011).

➡ Um formalismo operacional é dito **reconhecedor** no sentido que permite a “análise” de uma dada entrada para verificar se ela é **reconhecida** pela máquina.

Um pouco sobre sistemas de estados finitos...

Basicamente, um **sistema de estados finitos** é um modelo matemático de sistema com **entradas e saídas discretas** (em oposição ao contínuo) (Menezes 2011).
Características de tais sistemas:

- Assumir um **número finito e predefinido de estados**.
  - Assim, todos os estados do sistema podem ser explicitados antes de iniciar o processamento.
- Cada estado resume informações do passado para determinar as ações para a próxima entrada.

Sistemas de estados finitos: exemplos

Sistemas de estados finitos podem ser associados a diversos tipos de sistemas **naturais** e **construídos**.

Exemplo clássico: elevador

- Não memoriza as requisições anteriores;
- Cada "estado" sumariza as informações:
  - "Andar corrente";
  - "Direção de movimento".
- Entradas: requisições pendentes.

Sistemas de estados finitos: exemplos

Sistemas de estados finitos podem ser associados a diversos tipos de sistemas **naturais** e **construídos**.

Exemplo clássico: elevador

- Não memoriza as requisições anteriores;
- Cada "estado" sumariza as informações:
  - "Andar corrente";
  - "Direção de movimento".
- Entradas: requisições pendentes.

Contraexemplo: cérebro humano

- Composto por cerca de 2^{35} células (Menezes 2011).
- O elevado número de combinações (i.e., estados) torna essa abordagem pouco eficiente em termos práticos (explosão de estados).

O que é um autômato finito determinístico (AFD)?

Formalismo operacional ou reconhecedor muito importante em diversos estudos teórico-formais da computação.

Um AFD é um **sistema de estados finitos**, portanto, possui um **número finito e predefinido de estados** (Menezes 2002; Menezes 2011)

Por que determinístico?

Visto que para o estado corrente e o símbolo lido da entrada, tais autômatos sempre assumem **um único estado bem determinado**.

Visão geral de AFDs

Um AFD pode ser visto como uma máquina composta, basicamente, de **três partes** (Menezes 2002):

- **Fita:** que desempenha o papel de dispositivo de entrada
- **Unidade de controle:** reflete o estado da máquina. Possui uma cabeça de leitura que acessa uma célula da fita por vez e movimenta-se exclusivamente para a direita.
- **Função de transição:** função que comanda as leituras e dita o estado da máquina.

Sobre a fita. . .

Considerando o formalismo que será apresentado no decorrer desta disciplina, a fita é:

- **Finita** (à esquerda e à direita)
- **Dividida em células:** cada célula armazena um símbolo do alfabeto de entrada.
- **Não é possível gravar sobre a fita**.

···				1	1	0	1	1	0				···

Sobre a fita. . .

Considerando o formalismo que será apresentado no decorrer desta disciplina, a fita é:

- **Finita** (à esquerda e à direita)
- **Dividida em células:** cada célula armazena um símbolo do alfabeto de entrada.
- **Não é possível gravar sobre a fita**.

1	1	0	1	1	0

A unidade de controle e a cabeça de leitura...

Conforme mencionado, a unidade de controle possui um **número finito e predefinido de estados**.

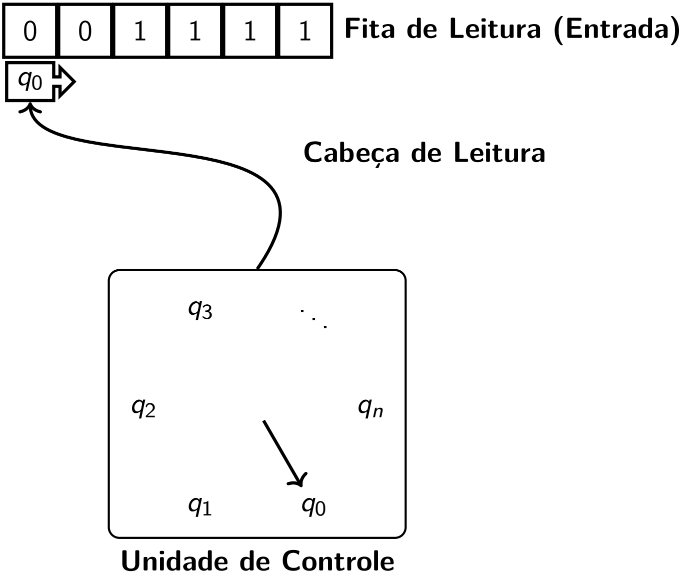
- A unidade de controle lê um símbolo da entrada (fita) por vez.

Após a leitura, a cabeça de leitura sempre se move uma célula para a direita.

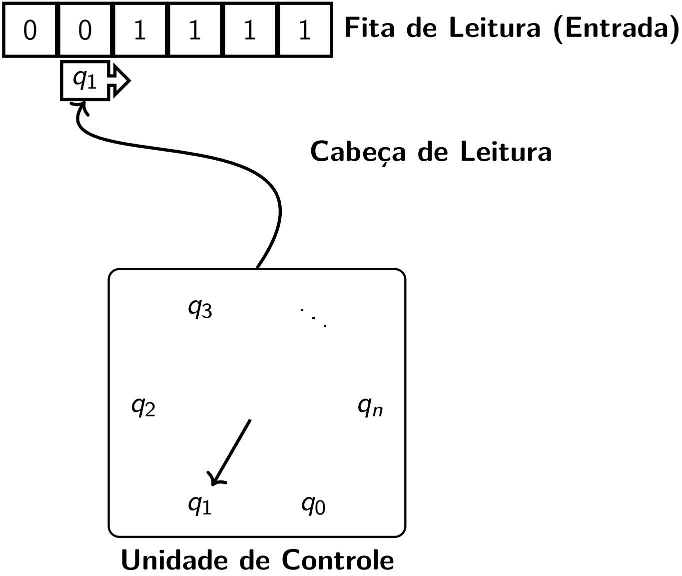
- Inicialmente, a cabeça de leitura encontra-se posicionada na célula mais à esquerda da fita.

☞ O **“programa”** é uma **função parcial** tal que: dependendo do ❶ estado corrente e do ❷ símbolo lido, **determina o novo estado do autômato**.

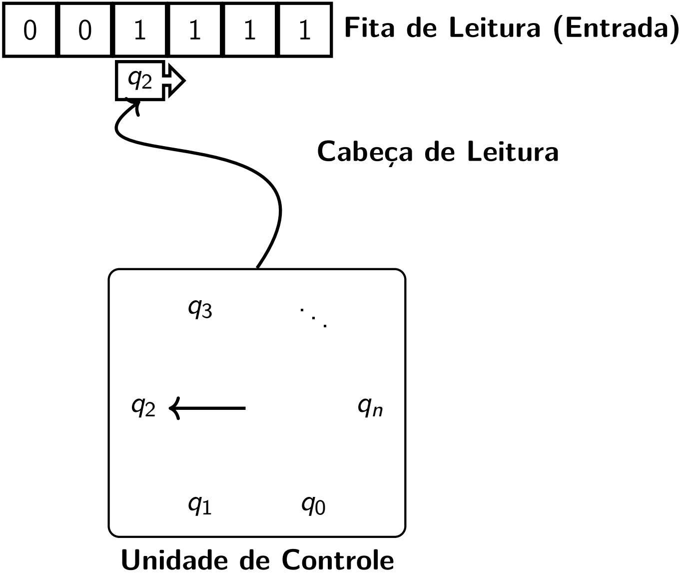
Visão geral: fita, cabeça de leitura, unidade de controle e programa.



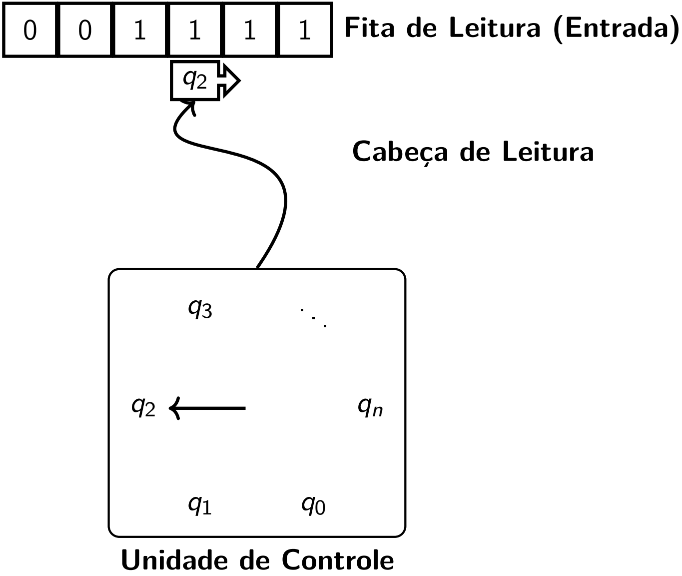
Visão geral: fita, cabeça de leitura, unidade de controle e programa.



Visão geral: fita, cabeça de leitura, unidade de controle e programa.



Visão geral: fita, cabeça de leitura, unidade de controle e programa.



Definição formal

Definição → *Autômato Finito Determinístico*

Um AFD $\mathcal{M}$ é uma quíntupla: $\mathcal{M} = (\Sigma, Q, \delta, q_0, F)$ onde:

- Σ representa o alfabeto de símbolos de entrada;
- Q é o conjunto finito de estados do autômato;
- δ função de transição ($\delta : Q \times \Sigma \to Q$) – a qual é uma função parcial;
  - Supondo que a função programa é definida para um estado p e um símbolo a resultando no estado q, então:

    $$\delta(p, a) = q$$

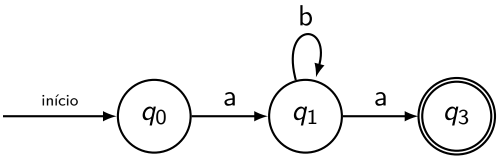
    é uma transição do AFD.
- $q_0 \in Q$ estado inicial;
- $F \subset Q$ representa o conjunto de estados finais.

❑

Representando AFDs: diagrama/grafo (1)

AFDs podem ser representados na forma de diagramas nos quais:

- Estados são nós/vértices, representados por círculos;
- Transições são arestas, ligando os vértices;
- Estados iniciais e finais são representados de forma distinta.



Representando AFDs: tabela (2)

Uma forma alternativa de se representar a função programa é por meio de uma **tabela de dupla entrada**.

Por exemplo, uma transição do tipo $\delta(p, a) = q$ pode ser representada como abaixo:

δ	a	...
p	q	...
q	...	...

Exemplo: um AFD trivial (1)...

Considere o AFD que aceita a linguagem L_1 descrita formalmente como:

$$L_1 = \{w \mid \text{w possui } aa \text{ ou } bb \text{ como subpalavras}\}$$

Exemplo: um AFD trivial (1)...

Considere o AFD que aceita a linguagem L_1 descrita formalmente como:

$$L_1 = \{w \mid \text{w possui } aa \text{ ou } bb \text{ como subpalavras}\}$$

- Alfabeto?

Exemplo: um AFD trivial (1)...

Considere o AFD que aceita a linguagem L_1 descrita formalmente como:

$$L_1 = \{w \mid \text{w possui } aa \text{ ou } bb \text{ como subpalavras}\}$$

- Alfabeto?
  - $\Sigma = \{a, b\}$
- Exemplos de palavras aceitas pelo AFD?

Exemplo: um AFD trivial (1)...

Considere o AFD que aceita a linguagem L_1 descrita formalmente como:

$$L_1 = \{w \mid \text{w possui } aa \text{ ou } bb \text{ como subpalavras}\}$$

- Alfabeto?
  - $\Sigma = \{a, b\}$
- Exemplos de palavras aceitas pelo AFD?
  - aa;
  - bb;
  - $abbab$; e
  - $aabbbb$.
- Exemplos de palavras rejeitadas pelo AFD?

Exemplo: um AFD trivial (1)...

Considere o AFD que aceita a linguagem L_1 descrita formalmente como:

$$L_1 = \{w \mid \text{w possui } aa \text{ ou } bb \text{ como subpalavras}\}$$

- Alfabeto?
  - $\Sigma = \{a, b\}$
- Exemplos de palavras aceitas pelo AFD?
  - aa;
  - bb;
  - $abbab$; e
  - $aabbbb$.
- Exemplos de palavras rejeitadas pelo AFD?
  - ε;
  - a; e
  - $ababab$.

Exemplo: um AFD trivial (2)...

Formalmente, o AFD que aceita L_1 pode ser descrito como:

$$M_1 = (\{a, b\}, \{q_0, q_1, q_2, q_f\}, \delta_1, q_0, q_f\}$$

onde δ_1 é dado pela tabela abaixo:

Exemplo: um AFD trivial (2)...

Formalmente, o AFD que aceita L_1 pode ser descrito como:

$$M_1 = (\{a, b\}, \{q_0, q_1, q_2, q_f\}, \delta_1, q_0, q_f)$$

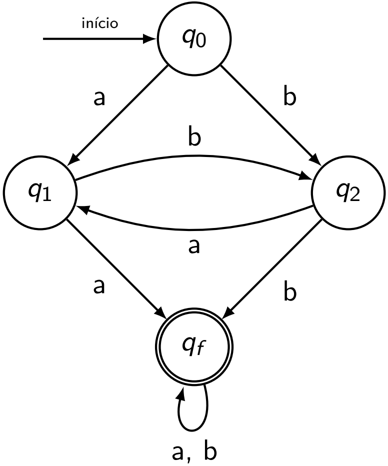
onde δ_1 é dado pela tabela abaixo:

δ_1	a	b
q_0	q_1	q_2
q_1	q_f	q_2
q_2	q_1	q_f
q_f	q_f	q_f

Exemplo: um AFD trivial (3)...

O AFD M_1 pode ser representado em forma de diagrama conforme mostrado ao lado.

- O AFD usa os estados q_1 e q_2 para "memorizar" o símbolo lido anteriormente. Assim:
  - q_1: "símbolo anterior foi a"
  - q_2: "símbolo anterior foi b"
- Qual informação memorizada por q_0 e q_f?



Exercícios (1)

Considerando o alfabeto $\Sigma = \{1, 0\}$, crie AFDs que aceitam as linguagens descritas a seguir.

Exercício ①: $L_{e1} = \{w \ : \ \text{w possui 01 como subpalavra}\}$

Exercício ②: $L_{e2} = \{w \ : \ \text{w possui um número par de } \mathit{0} \text{ e um número par de } \mathit{1}\}$

Exercício ③: $L_{e3} = \{w \ : \ \text{w possui pelo menos três 0}\}$

Exercícios (2): agora vamos tentar resolver algo ligeiramente mais complexo...

Considerando o alfabeto $\Sigma = \{a, b, c\}$, crie um AFD que aceita a linguagem descrita a seguir.

Exercício ④: $L_{e4} = \{w \ : \ \text{w possui aab como subpalavra}\}$

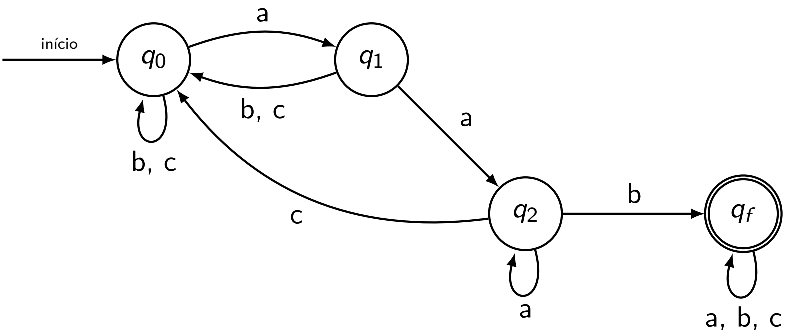
✍ **Solução**:

Exercícios (2): agora vamos tentar resolver algo ligeiramente mais complexo...

Considerando o alfabeto $\Sigma = \{a, b, c\}$, crie um AFD que aceita a linguagem descrita a seguir.

Exercício ④: $L_{e4} = \{w \;\; : \;\; \text{w possui aab como subpalavra}\}$

✍ **Solução**:



Como um AFD processa palavras?

☞ A linguagem aceita por um AFD é o conjunto de todas as palavras **"aceitas"** pelo AFD.

➡ Como saber se um AFD aceita uma palavra $a_1 a_2 \cdots a_n$?

Como um AFD processa palavras?

☞ A linguagem aceita por um AFD é o conjunto de todas as palavras **"aceitas"** pelo AFD.

➠ Como saber se um AFD aceita uma palavra $a_1 a_2 \cdots a_n$?

- O autômato começa o processamento no estado inicial, q_0.

Como um AFD processa palavras?

☞ A linguagem aceita por um AFD é o conjunto de todas as palavras **"aceitas"** pelo AFD.

➟ Como saber se um AFD aceita uma palavra $a_1 a_2 \cdots a_n$?

- O autômato começa o processamento no estado inicial, q_0.
- Inicialmente, consulta-se a função de transição δ para o primeiro símbolo da palavra: digamos $\delta(q_0, a_1) = q_1$.

Como um AFD processa palavras?

☞ A linguagem aceita por um AFD é o conjunto de todas as palavras **"aceitas"** pelo AFD.

➡ Como saber se um AFD aceita uma palavra $a_1 a_2 \cdots a_n$?

- O autômato começa o processamento no estado inicial, q_0.
- Inicialmente, consulta-se a função de transição δ para o primeiro símbolo da palavra: digamos $\delta(q_0, a_1) = q_1$.
- Em seguida, processa-se o próximo símbolo da palavra, a_2, avaliando $\delta(q_1, a_2) = q_2$

Como um AFD processa palavras?

☞ A linguagem aceita por um AFD é o conjunto de todas as palavras **"aceitas"** pelo AFD.

➡ Como saber se um AFD aceita uma palavra $a_1 a_2 \cdots a_n$?

- O autômato começa o processamento no estado inicial, q_0.
- Inicialmente, consulta-se a função de transição δ para o primeiro símbolo da palavra: digamos $\delta(q_0, a_1) = q_1$.
- Em seguida, processa-se o próximo símbolo da palavra, a_2, avaliando $\delta(q_1, a_2) = q_2$
- O processamento continua encontrando estados $q_3, q_4, \cdots, q_n$ de forma que $\delta(q_{i-1}, a_i)$ para cada i.

Como um AFD processa palavras?

☞ A linguagem aceita por um AFD é o conjunto de todas as palavras **"aceitas"** pelo AFD.

➡ Como saber se um AFD aceita uma palavra $a_1 a_2 \cdots a_n$?

- O autômato começa o processamento no estado inicial, q_0.
- Inicialmente, consulta-se a função de transição δ para o primeiro símbolo da palavra: digamos $\delta(q_0, a_1) = q_1$.
- Em seguida, processa-se o próximo símbolo da palavra, a_2, avaliando $\delta(q_1, a_2) = q_2$
- O processamento continua encontrando estados $q_3, q_4, \cdots, q_n$ de forma que $\delta(q_{i-1}, a_i)$ para cada i.
- Ao final do processamento, se q_n é membro de F, então a palavra $a_1 a_2 \cdots a_n$ é aceita – caso contrário a palavra é **"rejeitada"**.

A computação de um AFD...

A **computação de um AFD** para uma palavra w consiste na **sucessiva aplicação da função programa** para cada símbolo de w **até ocorrer uma condição de parada**.

Condições de parada

- Aceita a entrada w: após processar o último símbolo da fita, o AFD **assume um estado final**.
- Rejeita a entrada w. Duas possibilidades:
  - após processar o último símbolo da fita, o AFD **assume um estado não final**; ou
  - ao longo do processamento de w, a **função programa é indefinida** para o argumento (estado corrente e símbolo lido da fita).

Função programa estendida. . .

Para definir formalmente o comportamento de um AFD, é preciso estender a definição da função programa.

Definição → *Função Programa Estendida*

Seja $\mathcal{M} = (\Sigma, Q, \delta, q_0, F)$ um AFD. A função programa estendida denotada por:

$$\delta^* = Q \times \Sigma^* \to Q$$

é a função programa estendida para palavras e pode ser indutivamente definida como a seguir:

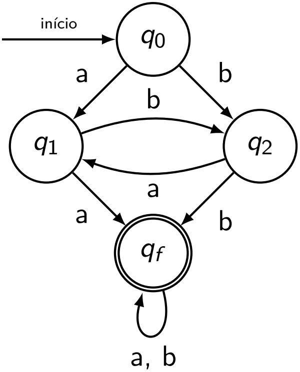
$$\begin{aligned} \delta^*(q, \varepsilon) &= q \\ \delta^*(q, aw) &= \delta^*(\delta(q, a), w) \end{aligned}$$

❑

Exemplo: função programa estendida

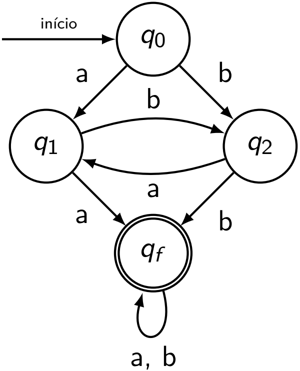
Dado $\mathcal{M}_1 = \{\{a, b\}, \{q_0, q_1, q_2, q_f\}, \delta_1, q_0, \{q_f\}\}$. A função de transição estendida aplicada à palavra *abaa* a partir do estado inicial q_0 é como segue:

$$\delta^*(q_0, abaa) = \delta^*(\delta(q_0, a), baa) =$$
$$\delta^*(q_1, baa) = \delta^*(\delta(q_1, b), aa) =$$
$$\delta^*(q_2, aa) = \delta^*(\delta(q_2, a)a) =$$
$$\delta^*(q_1, a) = \delta^*(\delta(q_1, a), \varepsilon) =$$
$$\delta^*(q_f, \varepsilon) = q_f$$



Exercício

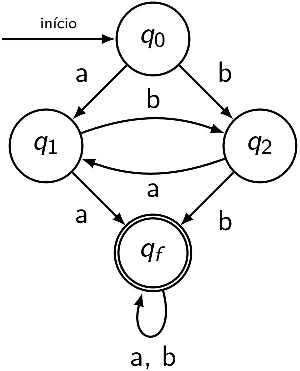
Exercício ⑤: Dado o AFD mostrado anteriormente: $\mathcal{M}_1 = \{\{a, b\}, \{q_0, q_1, q_2, q_f\}, \delta_1, q_0, \{q_f\}\}$. Detalhe a função de transição estendida aplicada à palavra *abab* a partir do estado inicial q_0.



✍ **Solução**:

Exercício

Exercício ⑤: Dado o AFD mostrado anteriormente: $\mathcal{M}_1 = \{\{a, b\}, \{q_0, q_1, q_2, q_f\}, \delta_1, q_0, \{q_f\}\}$. Detalhe a função de transição estendida aplicada à palavra *abab* a partir do estado inicial q_0.



✍ **Solução**:

$$\delta^*(q_0, abab) = \delta^*(\delta(q_0, a), bab) =$$
$$\delta^*(q_1, bab) = \delta^*(\delta(q_1, b), ab) =$$
$$\delta^*(q_2, ab) = \delta^*(\delta(q_2, a)b) =$$
$$\delta^*(q_1, b) = \delta^*(\delta(q_1, b), \varepsilon) =$$
$$\delta^*(q_2, \varepsilon) = q_2$$

Linguagem aceita

Seja $\mathcal{M} = \{\Sigma, Q, \delta, q_0, F\}$ um AFD, a linguagem **aceita** por $\mathcal{M}$ é denotada:

$$ACEITA(\mathcal{M}) \text{ ou } L(\mathcal{M})$$

é o conjunto de todas as palavras pertencentes a Σ^* aceitas por $\mathcal{M}$ a partir de q_0, ou seja:

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

Linguagem rejeitada

A linguagem **rejeitada** por $\mathcal{M}$ é denotada por:

$$REJEITA(\mathcal{M})$$

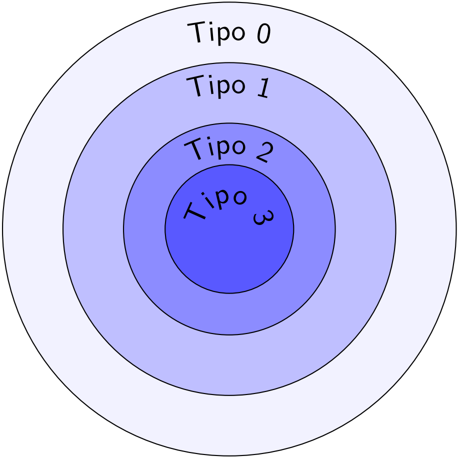
é o conjunto de todas as palavras pertencentes a Σ^* rejeitadas por $\mathcal{M}$ a partir de q_0, ou seja:

$$REJEITA(\mathcal{M}) = \{w \mid \delta^*(q_0, w) \notin F \text{ ou } \delta^*(q_0, w) \text{ é indefinida}\}$$

Como determinar se uma linguagem é regular?

Definição → *Linguagem Regular*

Uma linguagem L é dita regular (ou Tipo 3) se existe pelo menos um AFD que aceita L. ❑



Exercícios (3): agora vamos tentar resolver algo ainda mais complexo. . .

Considerando o alfabeto $\Sigma = \{0, 1\}$, crie um AFD que aceita a linguagem descrita a seguir.

Exercício ⑥: $L_{e4} = \{w \ : \ \text{w possui 1010 como subpalavra}\}$

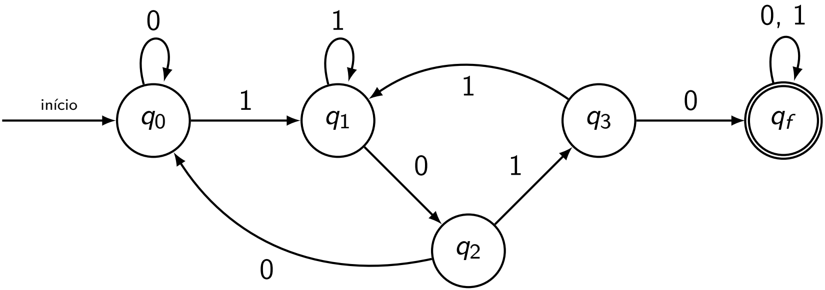
✍ **Solução**:

Exercícios (3): agora vamos tentar resolver algo ainda mais complexo. . .

Considerando o alfabeto $\Sigma = \{0, 1\}$, crie um AFD que aceita a linguagem descrita a seguir.

Exercício ⑥: $L_{e4} = \{w \ : \ \text{w possui 1010 como subpalavra}\}$

✍ **Solução**:



1. Linguagens regulares
2. Sistemas de estados finitos
3. Autômato finito determinístico
   - Fita
   - A unidade de controle e a cabeça de leitura
   - Definição formal
   - Representando AFDs
   - Exemplo
   - Processamento
4. Estendendo a função programa
5. **Considerações finais**

Considerações finais...

Na aula de hoje nós vimos:

- Autômatos finitos determinísticos;
  - tabela;
  - diagrama/grafo.
- Função de transição estendida.

Na **próxima aula**: autômatos finitos não-determinísticos.

Referências

Chomsky, N. (1956). "Three models for the description of language". In: *IRE Transactions on Information Theory* 2.3, pp. 113–124.

Menezes, Paulo Blauth (2002). *Linguagens Formais e Autômatos*. 3rd ed. Livros Didáticos do Instituto de Informática da UFRGS. Sagra Luzzatto, p. 165.

– (2011). *Linguagens Formais e Autômatos*. 6th ed. Livros Didáticos Informática da UFRGS. Bookman, p. 256.

☺**Próxima aula: exercício(s) sobre o conteúdo da aula de hoje!** ☺