Introdução à Teoria da Computação

Autômatos Finitos Não Determinísticos (NFA)

Professor Luís Carlos Pompeu

Autômato finto não determinístico

- Não-determinismo:
  - Importante generalização dos modelos de máquinas;
  - Fundamental em estudos:
    - Teoria da Computação,
    - Linguagens Formais,
    - Modelo para concorrência, ...

Autômato finto não determinístico

- O não determinismo não aumenta o poder de reconhecimento de linguagens da classe de autômatos.
  - Ou seja, qualquer linguagem reconhecida e aceita pelo autômato não determinístico é também aceita por autômatos determinísticos.
  - Qualquer autômato finito não determinístico pode ser **simulado** por um autômato finito determinístico.

Autômato finto não determinístico

- Não-determinismo no programa, é uma **função parcial**:
  - O **não-determinismo** em um programa, especialmente no contexto de autômatos, significa que, para uma mesma entrada e estado atual, o programa pode ter múltiplos caminhos possíveis. Em outras palavras, a execução não segue um único fluxo pré-definido, mas sim um conjunto de possibilidades.
  - "**É uma função parcial**" → Isso significa que a função de transição do autômato pode não estar definida para todas as combinações de estado e entrada. Ou seja, pode haver situações em que o programa não sabe exatamente para onde ir.

Autômato finto não determinístico

- **"Dependendo do estado corrente e do símbolo lido"**
  - O comportamento do programa depende do estado atual e do que ele está processando no momento.
- **"Determina um conjunto de estados do autômato"**
  - Diferente de um programa determinístico (onde sempre há um único próximo estado), aqui o programa pode escolher entre vários estados possíveis.
  - Um autômato não-determinístico pode escolher entre várias direções sem uma regra única para decidir.
  - O autômato pode, de fato, seguir todas as possibilidades ao mesmo tempo (conceito de computação paralela teórica).

Autômato finto não determinístico

- **Exemplo didático**: Caixa Eletrônico (Determinístico vs. Não-Determinístico)
- **Determinístico**: Você insere seu cartão e digita a senha. Se a senha estiver correta, você sempre acessa sua conta. Se errada, sempre dá erro.
- **Não-Determinístico**: Você insere seu cartão e digita a senha, mas o sistema pode:
  - Acessar sua conta normalmente
  - Solicitar verificação extra por SMS
  - Solicitar que você vá até um atendente
  - Travar o cartão se detectar algo suspeito
- Aqui, a mesma ação pode ter múltiplos resultados possíveis.

Autômato finto não determinístico

O AFN assume um conjunto de estados alternativos:

- Multiplicação da unidade de controle;
- Uma para cada alternativa;
- Unidades de Controle processando independentemente e sem compartilhar recursos

Autômato finto não determinístico

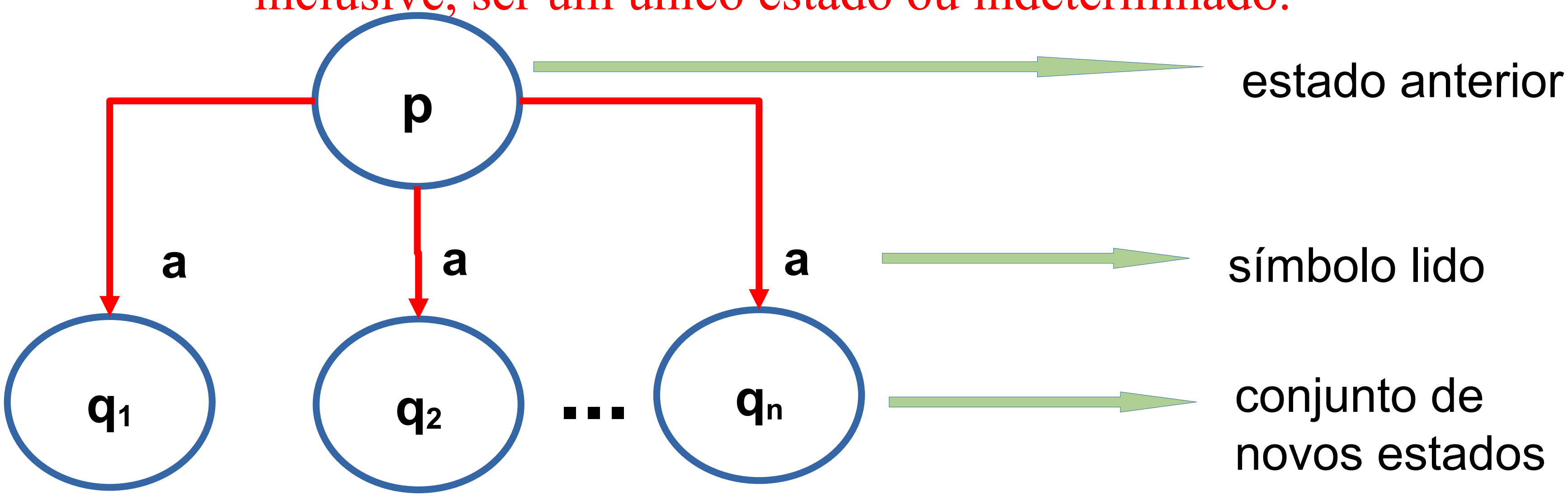
$M = (\Sigma, Q, \sigma, q_0, F)$

- **Σ**: Alfabeto (de símbolos) de entrada;
- **Q**: Um conjunto de estados possíveis do autômato (finito);
- **σ**: Programa ou Função de Transição (função parcial);
  - **$\sigma: Q \times \Sigma \rightarrow 2^Q$**
- Transição: **$\sigma(p, a) = \{q_1, q_2, \ldots, q_n\}$**
- **q_0**: Estado inicial (é um elemento distinguido de Q);
- **F**: Conjunto de estados finais (é um subconjunto de Q).

Autômato finto não determinístico

- Autômato como diagrama:

$\sigma(p,a) = \{q_1, q_2, \ldots, q_n\} \rightarrow$ o conjunto de estados pode, inclusive, ser um único estado ou indeterminado.



Autômato finto não determinístico

- **Computação** (Função Programa Estendida) de um autômato finito não-determinístico:
  - Sucessiva aplicação da função programa...
  - para cada símbolo da entrada (da esquerda para a direita)...
  - até ocorrer uma **condição de parada**.

- Argumentos para computação:
  - Conjunto finito de estados e uma palavra.

Autômato finto não determinístico

- $M = (\Sigma, Q, \sigma, q_0, F)$ autômato finito não-determinístico

  $$\sigma^*: 2^Q \times \Sigma^* \to 2^Q$$

- Indutivamente definida
  - **$\sigma^*(P, \varepsilon) = P$** (o estado **p** foi substituído por um conjunto de estados **P**)

    Se estou em um conjunto de estados e leio vazio, permaneço no mesmo conjunto de estados;
  - **$\sigma^*(P, aw) = \sigma^*(\cup_{q \in P} \sigma(q, a), w)$**

    Se estou em um conjunto de estados e leio um símbolo que é prefixo de uma palavra, vou transitar por um conjunto que é a união dos destinos de todos os resultados da função de transição para o símbolo lido
- Transição estendida
  - Para um conjunto de estados $\{q_1, q_2, \ldots, q_n\}$ e para o símbolo a:

    $\sigma^*(\{ q_1, q_2, \ldots, q_n \}, a) = \sigma(q_1, a) \cup \sigma(q_2, a) \cup \ \ldots \cup \sigma(q_n, a)$

Autômato finto não determinístico

- Parada do processamento:
  - Aceita a entrada:
    - Após processar o último símbolo da fita, ...
    - existe pelo menos **um estado final** ...
    - pertencente ao conjunto de estados alternativos atingidos.
- Rejeita a entrada. **Duas possibilidades**:
  - (1) Após processar o último símbolo da fita, todos os estados alternativos atingidos são **não finais**;
  - (2) Programa **indefinido** para o argumento (conjunto de estados e símbolo).
    - Exemplo: suponha que estou em determinado conjunto de estados e leio “a” e não tenho **nenhuma** transição para esse símbolo.

Linguagem Aceita, Linguagem Rejeitada

- Seja $M = (\Sigma, Q, \sigma, q_0, F)$ um AFN:

- Linguagem Aceita ou Linguagem Reconhecida por M
  - $L(M) = ACEITA(M) = \{ w \mid \sigma^*(\{ q_0 \}, w) \cap F \neq \varnothing \}$

    Toda a palavra "w" tal que o processamento da palavra toda, a partir de um estado inicial, a intersecção com um conjunto de estados finais "F" seja diferente de vazio

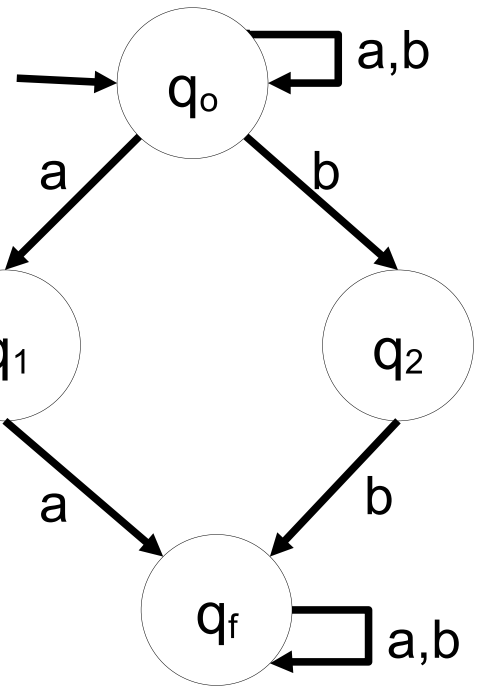
- Linguagem Rejeitada por M
  - $REJEITA(M) = \{w \mid \sigma^*(\{ q_0 \}, w) \cap F \neq \varnothing$ ou $\sigma^*(\{ q_0 \}, w)$ é indefinida $\}$

    Toda a palavra "w" tal que após o computação da palavra toda, a partir de um estado inicial, a intersecção com um conjunto de estados finais seja vazio ou em algum ponto da computação tenha todas as transições para o símbolo lindo como indefinido

Exemplo: aa ou bb como subpalavra

L_5 = { w | w possui aa ou bb como subpalavra }

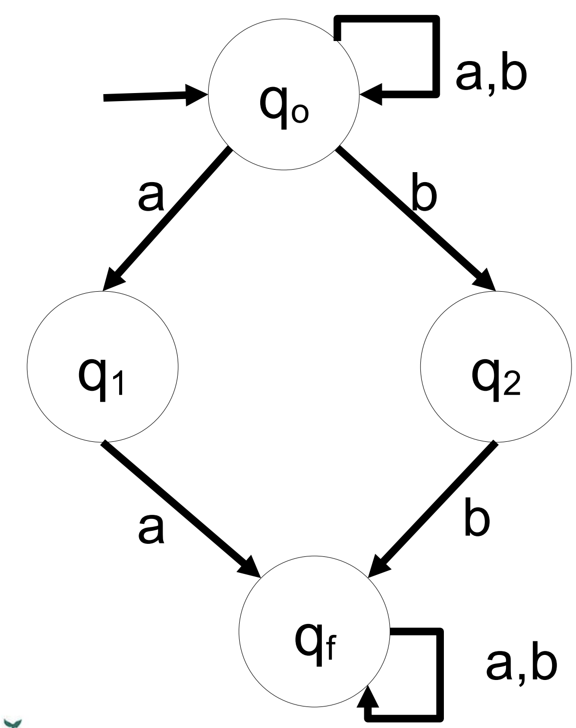
- Autômato finito não-determinístico:
  - $M_5 = (\{a, b\}, \{q_0, q_1, q_2, q_f\}, \sigma_5, q_0 \{q_f\})$



Considerando o estado inicial q0, se eu ler “a” vou para o estado q1, mas ao mesmo tempo, continuo em q0.
Isso pode ser comparado a *threads* em processamento paralelo, onde o processamento se divide em 2 partes cada parte segue caminhos independentes.
Se o próximo símbolo for “a”, saio de q1 e vou par qf, enquanto q0 continua em q0 e, nesse caso já atingi um estado de aceitação que é “aa” e já estou em um estado final. No entanto, o processamento continuará até o fim da palavra

Exemplo: aa ou bb como subpalavra

- O ciclo em q_0 realiza uma varredura em toda a entrada:
- O caminho $q^0/q_1/q_f$ garante a ocorrência de aa.
- O caminho $q_0/q_2/q_f$ garante a ocorrência de bb.



símbolos

Estados iniciais

σ_5	a	b
q0	{q0,q1}	{q0, q2}
q1	{qf}	-
q2	-	{qf}
qf	{qf}	{qf}

Transição indefinida

Transição indefinida

Exemplo: aa ou bb como subpalavra

- Computação da palavra abaa:
- $\sigma^*(\{q_0\}, abaa)$ = %Função estendida sobre **abaa**
- $\sigma^*(\sigma(q_0,a), baa)$ = %Processa **<u>a</u>baa**
- $\sigma^*(\{q_0,q_1\}, baa)$ = %Função estendida sobre **baa**
- $\sigma^*(\sigma(q_0,b) \cup \sigma(q_1,b), aa)$ = %Processa **<u>b</u>aa**
- $\sigma^*(\sigma(q_0,q_2) \cup \emptyset, aa)$ =
- $\sigma^*(\{q_0,q_2\}, aa)$ = %Função estendida sobre **aa**
- $\sigma^*(\sigma(q_0,a) \cup \sigma(q_2,a), a)$ = %Processa **<u>a</u>a**

Começar o vídeo aqui

Exemplo: aa ou bb como subpalavra

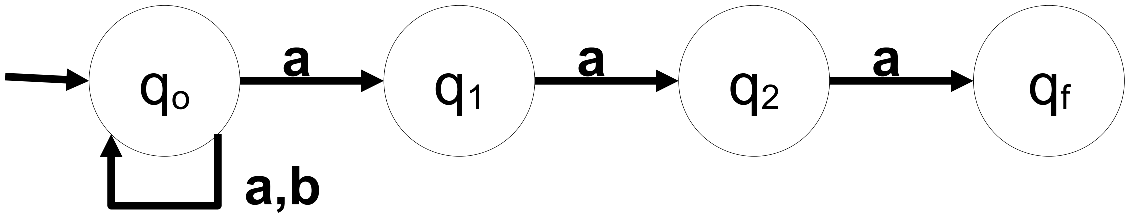
- Computação da palavra abaa: (continuação)
- $\sigma^*(\sigma(q_0,q_1) \cup \varnothing, a)$ =
- $\sigma^*(\{q_0,q_1\}, a)$ = %Função estendida sobre **a**
- $\sigma^*(\sigma(q_0,a) \cup \sigma(q_1,a), \varepsilon)$ = %Processa **a**
- $\sigma^*(\sigma(q_0,q_1) \cup \{q_f\}, \varepsilon)$ =
- $\sigma^*(\{q_0,q_1,q_f\}, \varepsilon)$ = $\{q_0,q_1,q_f\}$ %Função estendida sobre $\boldsymbol{\varepsilon}$
- **A palavra abaa é aceita pois {q0, q1, qf} ∩ F = $\{q_f\} \neq \varnothing$**

Exemplo: aaa como sufixo

$L_6 = \{ w \mid w \text{ possui aaa como sufixo} \}$

- Autômato finito não-determinístico:

$M_6 = (\{ a, b \}, \{ q_0, q_1, q_2, q_f \}, \sigma_6, q_0 \{ q_f \})$



Considere a palavra: abaabaaa

Exemplo: aaa como sufixo

Considere a palavra: abaabaaa

q0 leio **a** → q0, q1

q0 leio **b** → q0

q1 leio **b** → transição indefinida (-) sobra q0 para continuar

q0 leio **a** → q0, q1

q0 leio **a** → q0, q1

q1 leio **a** → q2 então aqui temos três estados q0, q1 e q2

q0 leio **b** → q0

q1 leio **b** → transição indefinida (-)

q2 leio **b** → transição indefinida (-) só sobrou o q0

q0 leio **a** → q0, q1

q0 leio **a** → q0, q1

q1 leio **a** → q2 novamente temos três estados q0, q1, q2

Exemplo: aaa como sufixo

Considere a palavra: abaabaaa

q0 leio **a** → q0, q1

q1 leio **a** → q2

q2 leio **a** → qf   temos 4 estados q0, q1, q2 e qf

q0 leio **Ɛ** → q0

q1 leio **Ɛ** → q1

q2 leio **Ɛ** → q2

qf leio **Ɛ** → qf

- A palavra é aceita porque temos 4 estados e pelo menos 1 deles é um estado final

Equivalência entre AFD e AFN

- Classe dos Autômatos Finitos Determinísticos:
  - É **equivalente** à classe dos Autômatos Finitos Não-Determinísticos.

- Não-determinismo:
  - Aparentemente, um significativo acréscimo ao poder computacional do autômato finito;
  - Na realidade não aumenta seu poder computacional.

Equivalência entre AFD e AFN

- Seja $\mathbf{M = (\Sigma, Q, \sigma, q_0, F)}$ um AFN qualquer
- E $\mathbf{M_D = (\Sigma, Q_D, \sigma_D, \langle q_0 \rangle, F_D)}$o AFD construído
- Q_D: todas as combinações, sem repetições, de estados de Q
  - Notação $\langle q_1\ q_2\ \ldots\ q_n \rangle$ é um estado determinístico que equivale as estados não determinísticos q1, q2, …, qn (separadamente)
  - A ordem não distingue combinações: $\langle q_u q_v \rangle = \langle q_v q_u \rangle$
- σ_D: $Q_D \times \Sigma \rightarrow Q_D$
- $\sigma_D(\langle q_1 \ldots q_n \rangle, a) = \langle p_1 \ldots p_m \rangle$ se $\sigma^*(\{q_1, \ldots, q_n\}, a) = \{p_1, \ldots, p_m\}$
  - $\langle q_0 \rangle$: Estado inicial
  - F_D: Conjunto de estados $\langle q_1\ q_2\ \ldots .qn \rangle$ pertencentes a Q_D:
    - Algum componente q_i pertence a F, para i em {1, 2, …, n}

Equivalência entre AFD e AFN

- Portanto, linguagem aceita por AFN:
  - É linguagem Regular ou Tipo 3

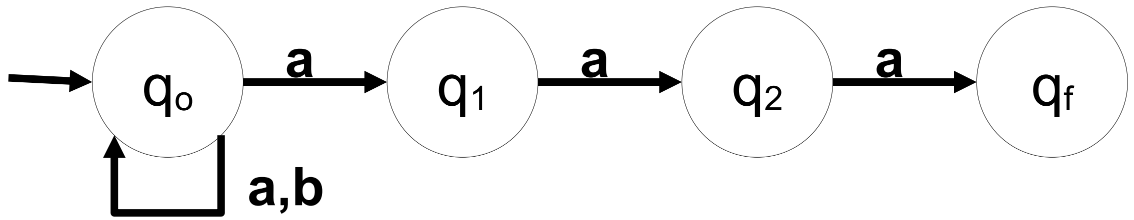
Determinismo X Não Determinismo

- Muitas vezes é mais fácil desenvolver um AFN do que um AFD.
  - Solução determinista:
    - **Não é trivial** - número grande de estados;
  - Solução não-determinista:
    - **Mais simples** poucos estados;

- Alternativa para construir um AFD:
  - **Desenvolver** inicialmente AFN;
  - **Converter** o AFN em AFD.

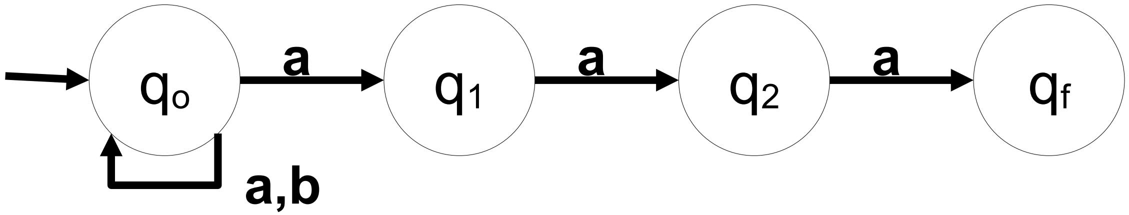
Exemplo: AFN → AFD

- $M_6 = (\{ a, b \}, \{ q_0, q_1, q_2, q_f \}, \sigma_6, q_0 \{ q_f \})$



Exemplo: AFN → AFD

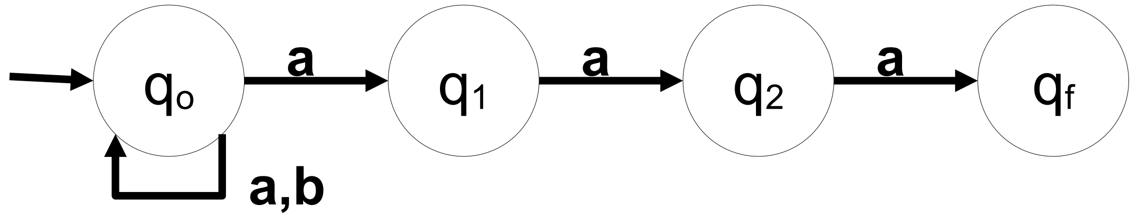
- $M_6 = (\{ a, b \}, \{ q_0, q_1, q_2, q_f \}, \sigma_6, q_0 \{ q_f \})$



- $M_6 = (\{ a, b \}, Q_D, \sigma_{6D}, <q_0>, F_D)$
- $Q_D = \{ <q_0>, <q1>, <q_2>, <q_f>, <q_0\ q_1>, <q_0\ q_2>,$
  - $<q_0\ q_f>, <q_1\ q_2>, \ldots, <q_0\ q_1\ q_2\ q_f> \}$
- Todas as combinações possíveis de estado de Q
- $F_D = \{ <q_f>, <q_0\ q_f>, <q_1\ q_f>, \ldots, <q_0\ q_1\ q_2\ q_f> \}$
- Subconjunto de Q, que possui pelo menos um qf

Exemplo: AFN → AFD

- AFN



σ_{6D}	a	b
$<q_0>$	$<q_0q_1>$	$<q_0>$
$<q_0q_1>$	$<q_0q_1q_2>$	$<q_0>$
$<q_0q_1q_2>$	$<q_0q_1q_2q_f>$	$<q_0>$
$<q_0q_1q_2q_f>$	$<q_0q_1q_2q_f>$	$<q_0>$

Quando o estado é indefinido, ignoro e mantenho apenas os definidos
Conversão de autômato não é inventar regra é seguir o algoritmo, ou seja, uma sequencia de passos que de um resultado exato

Exemplo: AFN → AFD

- AFD

σ_{6D}	a	b
$p_0 = <q_0>$	$<q_0q_1>$	$<q_0>$
$p_1 = <q_0q_1>$	$<q_0q_1q_2>$	$<q_0>$
$p_2 = <q_0q_1q_2>$	$<q_0q_1q_2q_f>$	$<q_0>$
$p_f = <q_0q_1q_2q_f>$	$<q_0q_1q_2q_f>$	$<q_0>$