Teoria de Linguagem

Autômatos Finitos Não Determinísticos

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

UFSJ
UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL-REI

Organização

Uma palavra sobre não determinismo. . .

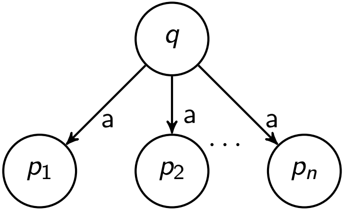
O não determinismo é uma importante generalização dos modelos de máquinas (Menezes 2011), sendo de fundamental importância no estudo das linguagens formais.

➟ A facilidade de não determinismo para autômatos pode ser descrita como:

> Dado o estado corrente e o símbolo lido da entrada, determina-se *aleatoriamente* um estado de um conjunto de estados alternativos.

Diferença...

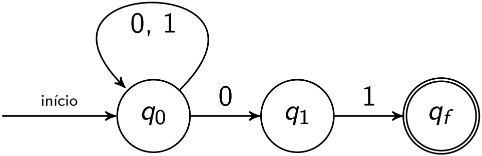
Essencialmente, a **principal diferença** entre um AFD e um autômato finito não determinístico (AFN) é que **o processamento de uma entrada** em um AFN **pode resultar em um conjunto de novos estados** (Menezes 2011).



- Em um AFD cada par *(estado, símbolo)* representa uma transição para um **único estado**.
- A cada transição não determinista, novos caminhos alternativos são possíveis, definindo-se assim uma **árvore de opções**.

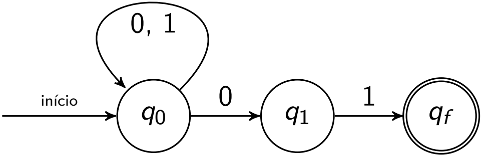
Exemplo (1)

Considere o AFN a seguir:



Exemplo (1)

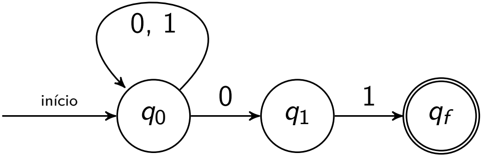
Considere o AFN a seguir:



- Qual a linguagem aceita pelo AFN?

Exemplo (1)

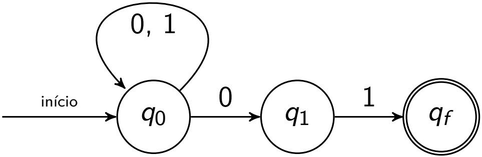
Considere o AFN a seguir:



- Qual a linguagem aceita pelo AFN?

$$L_e = \{w \mid \text{w possui } \mathit{01} \text{ como sufixo}\}$$

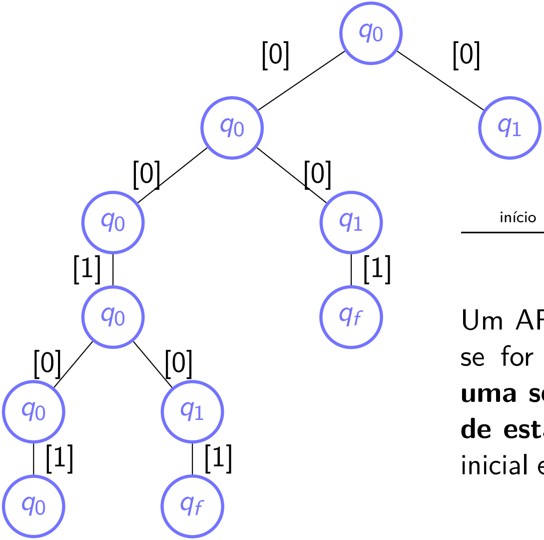
Exemplo (2)

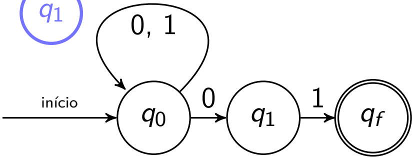


- Nota-se que o AFN permanece no estado q_0 (entre outros estados), enquanto ele não "adivinhou" que o sufixo 01 começou.
  - No estado q_0, para qualquer entrada, o autômato sempre assume o estado q_0 novamente (e o estado q_1 se a entrada for 0).
  - Quando a entrada é 0, o AFN assume tanto os estados q_0 e q_1.

Exemplo (3): AFN durante o processamento da palavra 00101

Árvore de opções:





Um AFN aceita uma palavra se for possível, a **partir de uma sequência de escolhas de estados**, partir do estado inicial e chegar no estado final.

Definição formal

Excetuando-se δ, os componentes Σ, Q, q_0, e F são como na definição do AFD.

Definição → *Autômato Finito Não Determinístico*

Um AFN $\mathcal{M}$ 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 ($\delta : Q \times \Sigma \to 2^Q$) – a qual é uma função total;
  - Assim, para um estado p e um símbolo a:

    $$\delta(p, a) = \{q_1, q_2, \ldots, q_n\}$$

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

Definição formal

Excetuando-se δ, os componentes Σ, Q, q_0, e F são como na definição do AFD.

Definição → *Autômato Finito Não Determinístico*

Um AFN $\mathcal{M}$ 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 ($\delta : Q \times \Sigma \rightarrow 2^Q$) – a qual é uma função total;
    - Assim, para um estado p e um símbolo a:

      $$\delta(p, a) = \{q_1, q_2, \ldots, q_n\}$$

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

Função programa estendida (1)...

Para definir formalmente o comportamento de um AFN, é necessário estender a definição da função programa usando como argumento um conjunto finito de estados e uma palavra.

Definição → *Função Programa Estendida (para AFNs)*

Seja $\mathcal{M} = (\Sigma, Q, \delta, q_0, F)$ um AFN, a função programa estendida de $\mathcal{M}$ pode ser denotada por:

$$\delta^* = 2^Q \times \Sigma^* \to 2^Q$$

sendo a função programa $\delta : Q \times \Sigma \to 2^Q$ estendida para palavras e indutivamente definida como segue:

$$\delta^*(P, \varepsilon) = P$$
$$\delta^*(P, aw) = \delta^*(\bigcup_{q \in P} \delta(q, a), w)$$

❑

Função programa estendida (2)...

A função programa estendida consiste na **sucessiva aplicação da função programa** a **cada símbolo da palavra**, a partir de **conjunto de estados**.

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

Condições de parada

- Aceita a entrada w: após processar o último símbolo da fita, **existe pelo menos um estado final** entre os estados alternativos atingidos.
- Rejeita a entrada w. Duas possibilidades:
  - após processar o último símbolo da fita, todos estados atingidos são não finais; ou
  - ao longo do processamento de w, o conjunto de estados alternativos é vazio. O autômato para por indefinição.

Exercícios (1)

Considerando o alfabeto $\Sigma = \{1, 0\}$:

Exercício ①: Crie um AFD que aceita a linguagem descrita a seguir.
$L_{e1} = \{w \;:\;$ w possui 1 na terceira posição do fim para o começo da palavra$\}$[a]

[a] e.g., 00100 pertence à L_{e1}, 0011 não.

Exercício ②: Crie um AFN que aceita L_{e1}.

Linguagem aceita

Seja $\mathcal{M} = \{\Sigma, Q, \delta, q_0, F\}$ um AFN, 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:

$$\boxed{L(\mathcal{M}) = \{w \mid \delta^*(\{q_0\}, w) \cap F \neq \emptyset\}}$$

Linguagem rejeitada

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

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

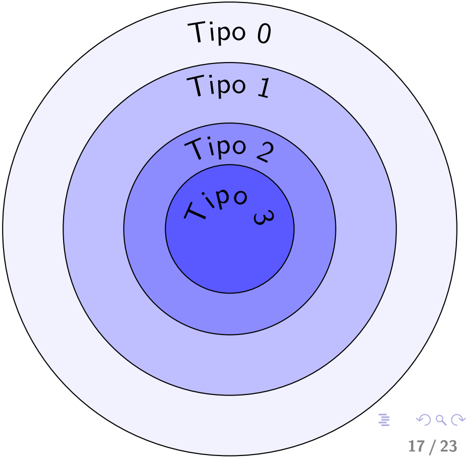
é o conjunto de todas as palavras pertencentes a Σ^* rejeitadas por $\mathcal{M}$ a partir de $\{q_0\}$, ou seja:

$$REJEITA(\mathcal{M}) = \{w \mid \delta^*(\{q_0\}, w) \cap F = \emptyset\}$$

Importante. . .

O não determinismo pode ser visto como uma **"facilidade"** que nem sempre aumenta o poder de reconhecimento dessa classe de autômatos.

- Importante **generalização** – então todo AFD é automaticamente um AFN (Sipser 2012).
- Conforme será mostrado, **qualquer AFN pode ser simulado por um AFD**.



Exercícios (2)

Considerando o alfabeto $\Sigma = \{a, b, c\}$:

Exercício ③: Construa um AFN que aceita palavras cujo último símbolo tenha aparecido anteriormente, e.g., *aba* é uma palavra aceita pelo AFN, *cacab* não é aceita.

Exercício ④: Considerando o AFN do exercício anterior, descreva computação da palavra *abbca* a partir do estado inicial do autômato (usando a função programa estendida, i.e. δ^*).

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

Considerando o alfabeto $\Sigma = \{1, 2, 3\}$, crie um AFN que aceita a linguagem descrita a seguir.

Exercício ⑤: $L = \{w$: tal que o último símbolo de w aparece pelo menos duas vezes, porém nenhum símbolo maior aparece entre as duas últimas ocorrências de tal símbolo.

☞ Exemplos de palavras que devem ser aceitas pelo AFN: 11, 2112, 123113, 3212113, etc.

Solução:

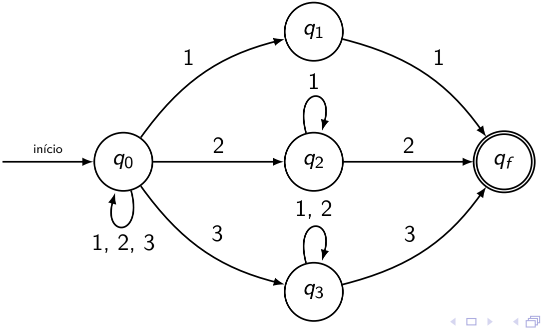
Exercícios (3): agora vamos tentar resolver algo ligeiramente mais complexo...

Considerando o alfabeto $\Sigma = \{1, 2, 3\}$, crie um AFN que aceita a linguagem descrita a seguir.

Exercício ⑤: $L = \{w\ :$ tal que o último símbolo de w aparece pelo menos duas vezes, porém nenhum símbolo maior aparece entre as duas últimas ocorrências de tal símbolo.

☞ Exemplos de palavras que devem ser aceitas pelo AFN: 11, 2112, 123113, 3212113, etc.

Solução:



Considerações finais...

Na aula de hoje nós vimos:

- Autômatos finitos não determinísticos (AFNs);
  - Diferenças entres AFDs e AFNs;
- Função de transição estendida (para AFNs).

Na **próxima aula**: equivalência entre AFDs e AFNs.

Referências

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

Sipser, Michael (2012). *Introduction to the Theory of Computation*. 3rd ed. Cengage Learning, p. 480.

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

Exercício Extra

✍ Considerando o alfabeto $\Sigma = \{1, 0\}$, crie um AFN que aceita palavras da seguinte forma:

- terminam em 01 e têm 011 como subpalavra; ou
- terminam em 10 e têm 100 como subpalavra.

Exercício Extra

✍ Considerando o alfabeto $\Sigma = \{1, 0\}$, crie um AFN que aceita palavras da seguinte forma:

- terminam em 01 e têm 011 como subpalavra; ou
- terminam em 10 e têm 100 como subpalavra.