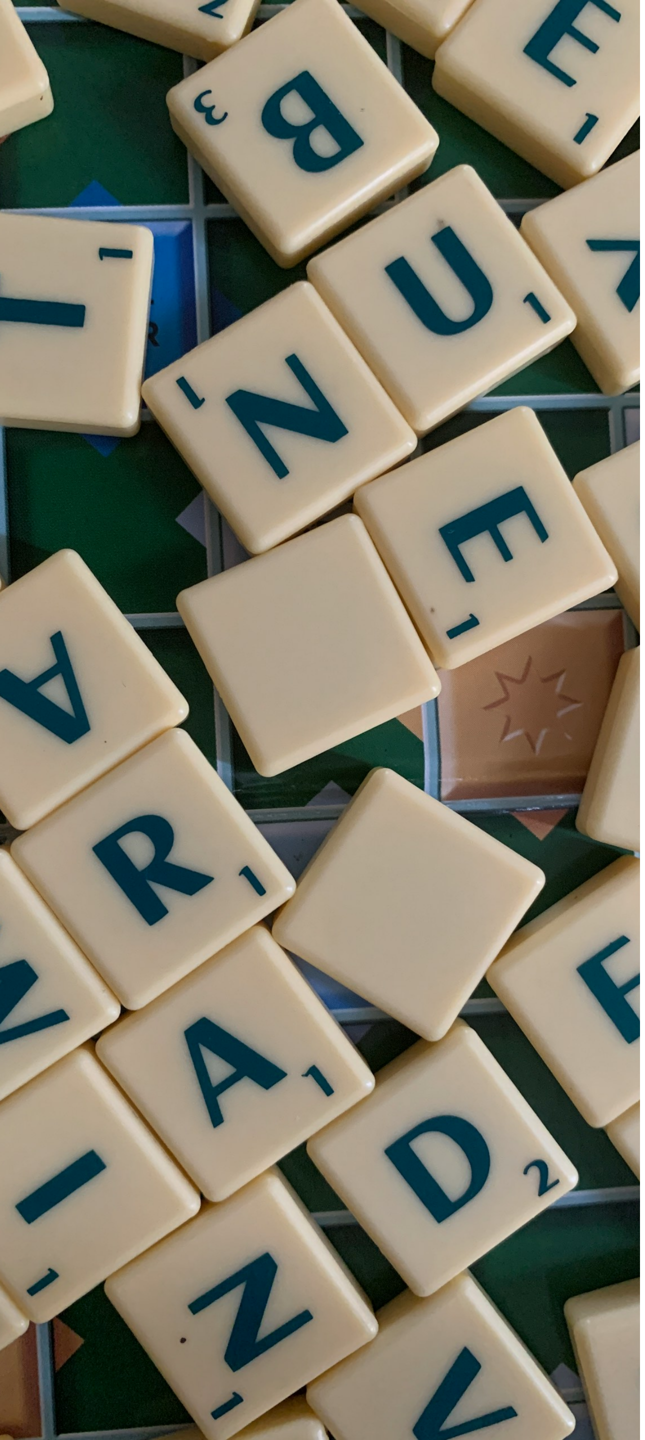


Autômatos Finitos Não-Determinísticos

Autômatos, Linguagens e Computação

Pontifícia Universidade Católica de Campinas

Prof. Dr. Denis M. L. Martins

Objetivos de Aprendizagem

- Compreender e formalizar a estrutura de um Autômato Finitos Não-Determinístico (AFN), identificando seus componentes
- Reconhecer as diferenças conceituais entre AFD e AFN.
- Transpor um AFN para seu equivalente AFD.

Não-Determinismo

- A máquina pode estar em **mais de um estado ao mesmo tempo**.
- **Motivação histórica**: A ideia de processamento paralelo.
- **Propriedade chave**: Existência de **caminhos** alternativos que podem levar à aceitação.

Exemplo de AFN na lousa

Transição ϵ

- Em determinado passo, a máquina pode escolher entre várias transições ou permanecer em estado **sem consumir símbolo**
- Todo AFN é um ϵ-AFN.

Definição Formal de AFN

Um AFN é uma 5-upla $\mathcal{A} = (Q,\ \Sigma,\ \delta,\ q_0,\ F)$ onde:

- Q conjunto finito de estados.
- Σ alfabeto finito.
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$.
- $q_0 \in Q$ estado inicial.
- $F \subseteq Q$ conjunto de estados finais.

Função de Transição Estendida em AFN

Definição formal: Dado um AFN $A = (Q, \Sigma, \delta, q_0, F)$, a função de transição estendida $\delta^* : Q \times \Sigma^* \to 2^Q$, definida como

$$\delta^*(q, w) = \begin{cases} \{q\} & \text{se } w = \varepsilon \quad \text{(palavra vazia)} \\ \delta(\delta^*(q, u), a) & \text{se } w = ua \text{ com } u \in \Sigma^*,\ a \in \Sigma \end{cases}$$

para, $q \in Q$ e $w \in \Sigma^*$.

Função de Transição Estendida em AFN (cont)

Podemos também estender a função de transição para conjuntos de estados $P \subseteq Q$ com $\delta^* : 2^Q \times \Sigma^*$:

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

Assim, temos que:

$$\delta^*(\{q_1, q_2, \ldots, q_r\}, w) = \delta^*(q_1, w) \cup \delta^*(q_2, w) \cup \ldots \cup \delta^*(q_r, w)$$

Ou seja, a união de todas as possibilidades de aplicação da função de transição sobre a cadeia w.

Aceite de cadeia em AFN

Dizemos que uma cadeia w é aceita por um AFN se existir **pelo menos uma** sequência de transições que leva a um estado de aceitação (final) ao processar todos os símbolos de w.

$$\delta^*(q_0, w) \cap F \neq \emptyset.$$

- Imagine que o AFN assume **múltiplas cópias** de si mesmo, seguindo todas as possibilidades em paralelo.
- "Always guesses right"

Linguagem de um AFN

A linguagem de um AFN $A = (Q, \Sigma, \delta, q_0, F)$ é

$$L(A) = \{w | \delta^*(q_0, w) \cap F \neq \emptyset\}$$

Não-determinismo é uma generalização do determinismo.

AFD e AFN são equivalentes

Equivalência

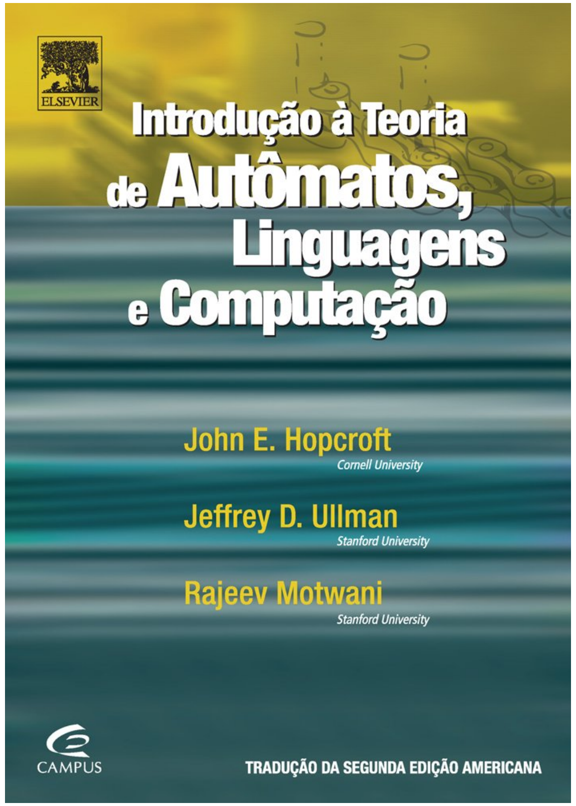
- Para todo AFN, existe um AFD que aceita sua linguagem.
  - Prova por subset construction.
- Número de estados de um AFD pode ser exponencial no número de estados de um AFN.
- A linguagem dos AFNs são linguagens regulares.

Exercícios

1. Mostre que a linguagem $L = \{\, w \in \{0,1\}^* \mid w \text{ tem número par de } 0s \,\}$ pode ser reconhecida por um AFN com apenas 2 estados.
2. Projete um AFN que reconheça a linguagem:
   $L = \{w \in \{0,1\}^* \mid w \text{ possui a subcadeia } 01\}$
3. Projete um AFN que reconheça a linguagem:
   $L = \{w \in \{0,1\}^* \mid |w|_1 \geq 1 \text{ e existe um num. par de 0s após o último } 1\}$

Resumo e Próximos Passos

- **Definição Formal**: AFN
  $A = (Q, \Sigma, \delta, q_0, F)$
- A máquina pode estar em **mais de um estado ao mesmo tempo**.
- Inspirado na ideia de processamento paralelo.
- Transições ϵ que permitem "pular" sem consumir símbolo.
- Equivalência com AFD (poderia ser reconhecida por um AFD).



Atividade recomendada: Leitura da seção 2.3 do capítulo 2.

Perguntas e Discussão

- Qual é a vantagem prática de usar um AFN em vez de converter imediatamente para um AFD?
- Em que situações o não-determinismo pode simplificar drasticamente a construção do autômato?
- Como a estrutura do AFN (por exemplo, presença de muitas transições ϵ) influencia o tamanho final do AFD?
- Quais linguagens não reconhecíveis por AFN?