LFA - Linguagens Formais e Autômatos

Aula 03 - Autômatos Finitos Não Determinísticos

Leandro Israel Pinto

UDESC

4 de Março de 2015

Sumário

1. Autômato Finito Não-Determinístico
   - Função Programa Estendida
2. AFD x AFN
3. Equivalência entre AFD e AFN
   - AFN→AFD
   - AFD→AFN

Sumário

1. Autômato Finito Não-Determinístico
   - Função Programa Estendida
2. AFD x AFN
3. Equivalência entre AFD e AFN
   - AFN→AFD
   - AFD→AFN

Autômato Finito Não-Determinístico (AFN) I

Definição

O Autômato Finito Não-Determinístico é uma 5-tupla definida como:

$$A = (\Sigma, Q, \delta, q_0, F)$$

Σ : Alfabeto;

Q : Conjunto finito de estados possíveis;

δ : função programa ou função de transição:

$$\delta : Q \times \Sigma \rightarrow 2^Q$$

q_0 : estado inicial, $q_0 \in Q$

F : conjunto de estados finais, $F \subseteq Q$

Autômato Finito Não-Determinístico (AFN) II

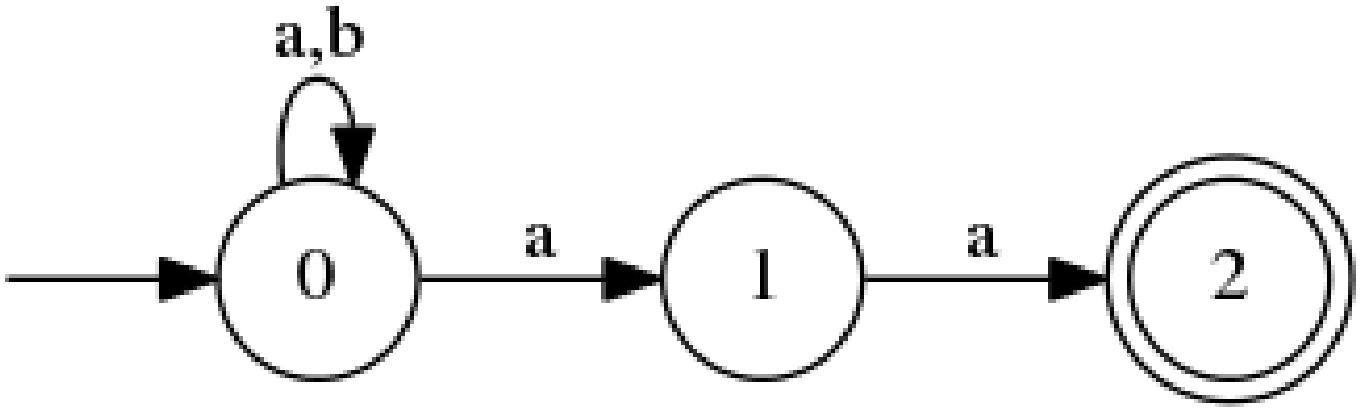
- A diferença de um AFN para um AFD está na função de transição δ;
- 2^Q significa todos os possíveis subconjuntos que podemos formar com os elementos de Q.
- Então se $Q = \{q0, q1\}$, $2^Q = \{\{q0\}, \{q1\}, \{q1, q2\}\}$;

Autômato Finito Não-Determinístico (AFN) III

- A diferença de um AFN para um AFD está na função de transição δ;
- 2^Q significa todos os possíveis subconjuntos que podemos formar com os elementos de Q.
- Então se $Q = \{q0, q1\}$, $2^Q = \{\{q0\}, \{q1\}, \{q1, q2\}\}$;
- Pode-se assumir mais de um estado ao mesmo tempo;
- Surgem mais unidades de controle, uma para cada caminho alternativo;
- Basta que uma unidade de controle alcance o estado final para que a palavra seja reconhecida;

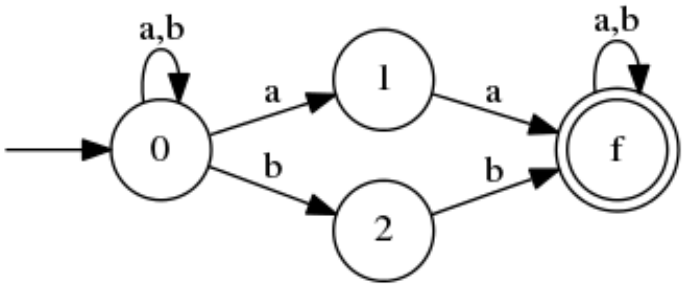
Autômato Finito Não-Determinístico (AFN) IV

Um exemplo:



Autômato Finito Não-Determinístico (AFN) V

Outro exemplo:



δ_1	a	b
q0	$\{q0, q1\}$	$\{q0, q2\}$
q1	$\{qf\}$	-
q2	-	$\{qf\}$
qf	$\{qf\}$	$\{qf\}$

$$A_1 = (\{a, b\}, \{q0, q1, q2, qf\}, \delta_1, q0, \{qf\})$$

Autômato Finito Não-Determinístico (AFN) VI

Linguagem Aceita/Rejeita

ACEITA(A) ou L(A) : Linguagem aceita pelo autômato A. É o conjunto de todas as palavras pertencentes a Σ^* tais que existe **pelo menos um** caminho alternativo que aceita a palavra. Assim:

$$ACEITA(A) = \{w | \ existe \ q \in \delta(q_0, w) \ tal \ que \ q \in F\}$$

REJEITA(A) : Todas as palavras pertencentes a Σ^* que são rejeitadas por A. Não há nenhum caminho que aceite a palavra;

Função Programa Estendida I

Definição

A função programa estendida é denotada por:

$$\underline{\delta} : 2^Q \times \Sigma^* \rightarrow 2^Q$$

$$\underline{\delta}(P, \varepsilon) = P \quad (1)$$

$$\underline{\delta}(P, aw) = \underline{\delta}(\cup_{q \in P} \delta(q, a), w) \quad (2)$$

Example

$$\underline{\delta}(\{q1, q2, ..., qn\}, a) = \delta(q1, a) \cup \delta(q2, a) \cup ... \cup \delta(qn, a)$$

Sumário

1 Autômato Finito Não-Determinístico
- Função Programa Estendida

2 AFD x AFN

3 Equivalência entre AFD e AFN
- AFN→AFD
- AFD→AFN

AFD x AFN

- AFN parece aumentar o poder computacional;
- Mas o poder computacional é o mesmo;
- Mesma capacidade de reconhecimento;

Para cada AFN é possível construir um AFD equivalente. O contrário também é verdadeiro.

Sumário

1. Autômato Finito Não-Determinístico
   - Função Programa Estendida
2. AFD x AFN
3. Equivalência entre AFD e AFN
   - AFN→AFD
   - AFD→AFN

Equivalência entre AFD e AFN I

Theorem

A classe dos AFD é equivalente a classe dos AFN.

Como provar?

- Mostrar que a partir de um AFN M qualquer, é possível construir um AFD M' que realiza o mesmo procedimento;
- O contrário, mostrar que a partir de um AFD qualquer pode-se construir um AFN equivalente, decorre trivialmente das definições $\neg\neg$.

AFN→AFD I

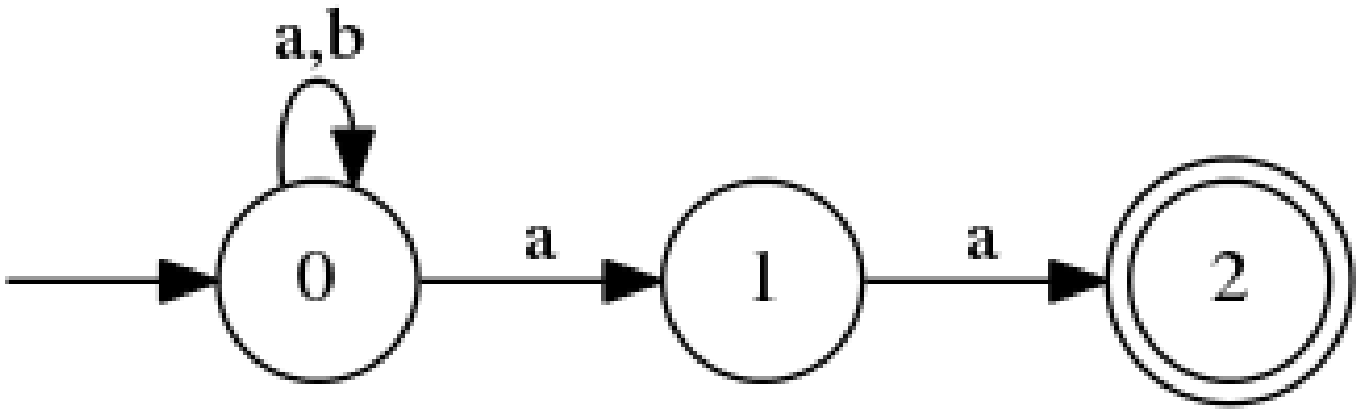
Como provar?

Seja $M = \{\Sigma, Q, \delta, q0, F\}$ um AFN qualquer. Seja $M' = \{\Sigma, Q', \delta, \langle q0 \rangle, F'\}$ um AFD construído a partir de M como segue:

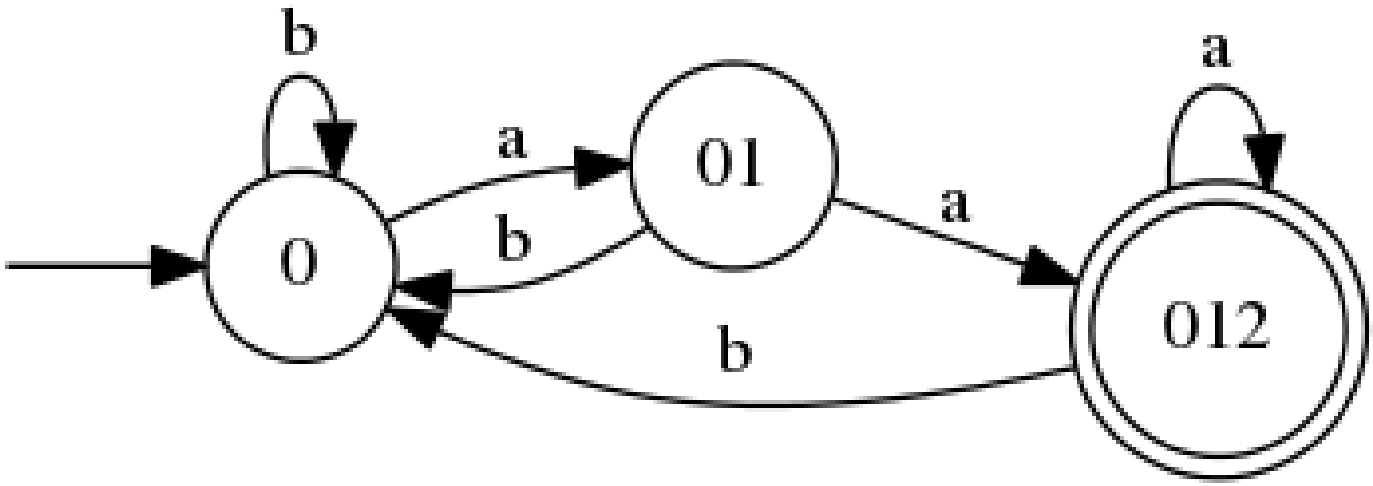
- Q' : Conjunto de combinações de estados de Q, denotado por $\langle q1q2...qn \rangle$;
- δ' : tal que $\delta(\langle q_1...q_n \rangle, a) = \langle p_1...p_m \rangle$ se, e somente se, $\delta(\{q_1, ..., q_n\}, a) = \{p_1, ..., p_m\}$;
- $\langle q_0 \rangle$: Estado inicial;
- F' : Conjunto de todos os estados $\langle q_1 q_2...q_n \rangle$ pertencentes a Q' tal que alguma componente q_i pertence a F;

AFN→AFD II

Converter o seguinte AFN em AFD:



AFN→AFD III



- Notem as combinações de estados (Q');
- Como é a função programa;
- Qual o nome do estado final;
- Compare com a prova anterior AFN para AFD;

AFD→AFN I

A demonstração de que o AFD M' simula o processamento do AFN M é por indução no tamanho da palavra.
Mostrar que

$$\delta'(\langle q_0 \rangle, w) = \langle q_1 ... q_u \rangle \quad sse \quad \delta(\{q_0\}, w) = \{q_1, ..., q_u\}$$

1. Base de indução. Seja w tal que $|w| = 0$. Então:

$$\delta'(\langle q_0 \rangle, \varepsilon) = \langle q_0 \rangle \quad sse \quad \delta(\{q_0\}, \varepsilon) = \{q_0\}$$

o que é verdadeiro, por definição de função programa estendida;

AFD→AFN II

2. Hipótese de indução. Seja w tal que $|w| = n$ e $n > 1$. Suponha verdadeiro que:

$$\delta'(\langle q_0 \rangle, w) = \langle q_1 ... q_u \rangle \quad sse \quad \delta(\{q_0\}, w) = \{q_1, ..., q_u\}$$

3. Passo de Indução. Seja w tal que $|wa| = n + 1$ e $n \geqslant 1$.

$$\delta'(\langle q_0 \rangle, wa) = \langle p_1 ... p_v \rangle \quad sse \quad \delta(\{q_0\}, wa) = \{p_1, ..., p_v\}$$

o que equivale, por hipótese de indução:

$$\delta'(\langle q_1 ... q_u \rangle, a) = \langle p_1 ... p_v \rangle \quad sse \quad \delta(\{q_1, ..., q_u\}, a) = \{p_1, ..., p_v\}$$

o que é verdadeiro, por definição de δ'.

References I