Teoria de Linguagem

Equivalência entre AFDs e AFNs

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

UFSJ
UNIVERSIDADE FEDERAL
DE SÃO JOÃO DEL-REI

Organização

1. Equivalência entre AFDs e AFNs

2. Demonstrando a equivalência entre AFDs e AFNs
   - Prova por construção
   - Prova por indução

3. Considerações finais

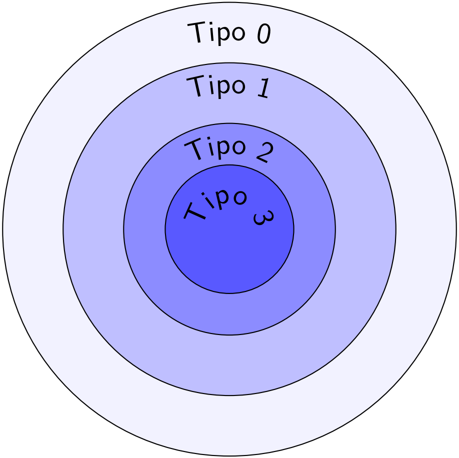
Então quer dizer que AFNs e AFDs são equivalentes?

Conforme mencionado na aula passada, a classe dos AFNs é equivalente à classe dos AFDs.

- O não determinismo **NÃO** aumenta o poder computacional dos autômatos (Sipser 2012).
- Conforme será mostrado, para cada AFN, é possível construir um AFD equivalente que realiza as mesmas computações.

Teorema

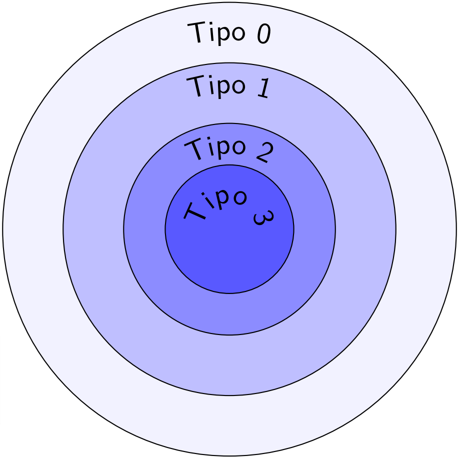
A classes dos AFNs é equivalente à classe dos AFDs.



Então quer dizer que AFNs e AFDs são equivalentes?

Conforme mencionado na aula passada, a classe dos AFNs é equivalente à classe dos AFDs.

- O não determinismo **NÃO** aumenta o poder computacional dos autômatos (Sipser 2012).
- Conforme será mostrado, para cada AFN, é possível construir um AFD equivalente que realiza as mesmas computações.



Teorema

A classes dos AFNs é equivalente à classe dos AFDs.

1. Equivalência entre AFDs e AFNs

2. Demonstrando a equivalência entre AFDs e AFNs
   - Prova por construção
   - Prova por indução

3. Considerações finais

Prova (1)

Construção de subconjuntos

A prova a seguir envolve a *"construção de subconjuntos"* (Hopcroft et al. 2006): construção de subconjuntos a partir do conjunto de estados do AFN.

✍ A construção do subconjunto começa com um AFN qualquer, $\mathcal{M}_N = (\Sigma, Q_N, \delta_N, q_0, F_N)$. A meta é construir um AFD $\mathcal{M}_D = (\Sigma, Q_D, \delta_D, \{q_0\}, F_D)$ tal que $L(\mathcal{M}_D) = L(\mathcal{M}_N)$.

Detalhes importantes

- O alfabeto dos dois autômatos é o mesmo.
- O estado inicial de $\mathcal{M}_D$ é o conjunto contendo somente o estado inicial de $\mathcal{M}_N$.

- Essencialmente, a ideia é construir estados para $\mathcal{M}_D$ que simulem as combinações de estados alternativos de $\mathcal{M}_N$ (Menezes 2011).

✍ Os elementos de $\mathcal{M}_D$ são construídos como a seguir:

- Q_D é o conjunto de todos os subconjuntos de Q_N, i.e., conjunto das partes de Q_N.[1]
- F_D é o conjunto de subconjuntos S pertencentes a Q_N tal que $S \cap F_N \neq \emptyset$.
- Para cada conjunto $S \subseteq Q_N$ e cada símbolo de entrada $a \in \Sigma$:

$$\delta_D(S, a) = \bigcup_{p \in S} \delta_N(p, a)$$

[1] Note que se Q_N tem n estados, então Q_D terá 2^n estados.

✍ Os elementos de $\mathcal{M}_D$ são construídos como a seguir:

- Q_D é o conjunto de todos os subconjuntos de Q_N, i.e., conjunto das partes de Q_N.[1]
- F_D é o conjunto de subconjuntos S pertencentes a Q_N tal que $S \cap F_N \neq \emptyset$.
- Para cada conjunto $S \subseteq Q_N$ e cada símbolo de entrada $a \in \Sigma$:

$$\delta_D(S, a) = \bigcup_{p \in S} \delta_N(p, a)$$

✍ Para computar $\delta_D(S, a)$:

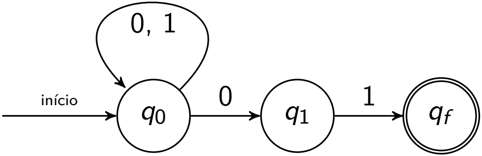
- É preciso analisar $\forall p \in S$ e verificar para quais estados o autômato $\mathcal{M}_N$ vai quando ele está em p e processa a.
- Em seguida, calcular a união dos estados.

[1] Note que se Q_N tem n estados, então Q_D terá 2^n estados.

Prova (3)

Demonstração construtiva

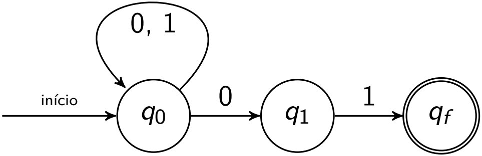
Considere o AFN a seguir:



Prova (3)

Demonstração construtiva

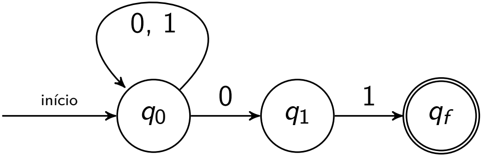
Considere o AFN a seguir:



- Qual a linguagem aceita pelo AFN?

Prova (3)
Demonstração construtiva

Considere o AFN a seguir:



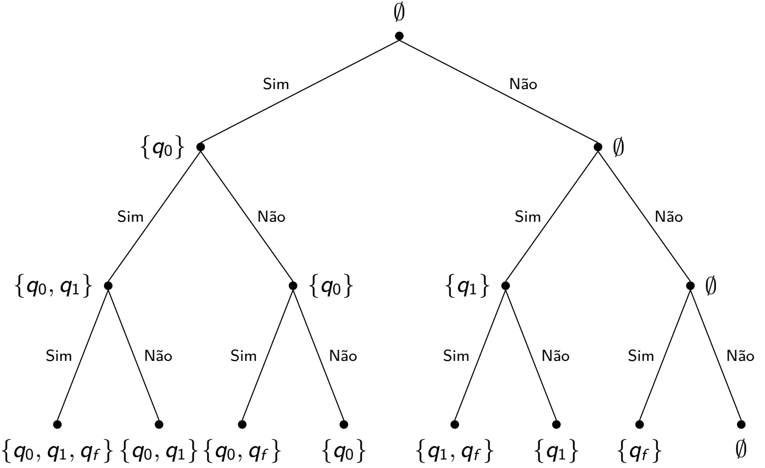
- Qual a linguagem aceita pelo AFN?

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

Prova (4)

Demonstração construtiva: calculando o conjunto das partes

Conforme mencionado, Q_D é o conjunto de todos os subconjuntos de Q_N:



Computando δ_D:

δ	0	1
$\emptyset$	$\emptyset$	$\emptyset$

Computando δ_D:

δ	0	1
$\emptyset$	$\emptyset$	$\emptyset$
$\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$

Computando δ_D:

δ	0	1
$\emptyset$	$\emptyset$	$\emptyset$
$\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_1\}$	$\emptyset$	$\{q_f\}$

Computando δ_D:

δ	0	1
$\emptyset$	$\emptyset$	$\emptyset$
$\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_1\}$	$\emptyset$	$\{q_f\}$
$*\{q_f\}$	$\emptyset$	$\emptyset$

Computando δ_D:

δ	0	1
$\emptyset$	$\emptyset$	$\emptyset$
$\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_1\}$	$\emptyset$	$\{q_f\}$
$*\{q_f\}$	$\emptyset$	$\emptyset$
$\{q_0, q1\}$	$\{q_0, q_1\}$	$\{q_0, q_f\}$

Computando δ_D:

δ	0	1
$\emptyset$	$\emptyset$	$\emptyset$
$\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_1\}$	$\emptyset$	$\{q_f\}$
$*\{q_f\}$	$\emptyset$	$\emptyset$
$\{q_0, q1\}$	$\{q_0, q_1\}$	$\{q_0, q_f\}$
$*\{q_0, q_f\}$	$\{q_0, q_1\}$	$\{q_0\}$

Computando δ_D:

δ	0	1
$\emptyset$	$\emptyset$	$\emptyset$
$\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_1\}$	$\emptyset$	$\{q_f\}$
$*\{q_f\}$	$\emptyset$	$\emptyset$
$\{q_0, q1\}$	$\{q_0, q_1\}$	$\{q_0, q_f\}$
$*\{q_0, q_f\}$	$\{q_0, q_1\}$	$\{q_0\}$
$*\{q_1, q_f\}$	$\emptyset$	$\{q_f\}$

Computando δ_D:

δ	0	1
$\emptyset$	$\emptyset$	$\emptyset$
$\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_1\}$	$\emptyset$	$\{q_f\}$
$*\{q_f\}$	$\emptyset$	$\emptyset$
$\{q_0, q1\}$	$\{q_0, q_1\}$	$\{q_0, q_f\}$
$*\{q_0, q_f\}$	$\{q_0, q_1\}$	$\{q_0\}$
$*\{q_1, q_f\}$	$\emptyset$	$\{q_f\}$
$*\{q_0, q_1, q_f\}$	$\{q_0, q_1\}$	$\{q_0, q_f\}$

Computando δ_D:

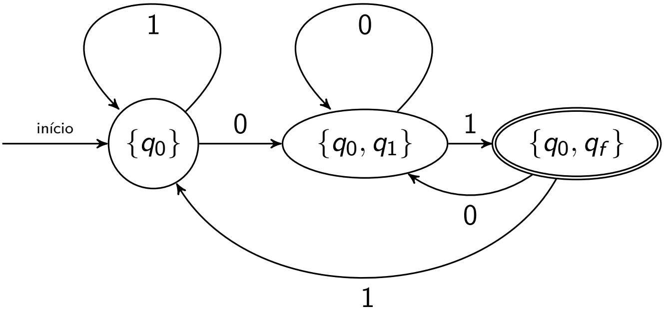
δ	0	1
$\emptyset$	$\emptyset$	$\emptyset$
$\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_1\}$	$\emptyset$	$\{q_f\}$
$*\{q_f\}$	$\emptyset$	$\emptyset$
$\{q_0, q1\}$	$\{q_0, q_1\}$	$\{q_0, q_f\}$
$*\{q_0, q_f\}$	$\{q_0, q_1\}$	$\{q_0\}$
$*\{q_1, q_f\}$	$\emptyset$	$\{q_f\}$
$*\{q_0, q_1, q_f\}$	$\{q_0, q_1\}$	$\{q_0, q_f\}$

Note que a função programa na forma de tabela pertence a um AFD (embora os estados sejam conjuntos).

Prova (6)

Demonstração construtiva

Iniciando em $\{q_0\}$, **somente 3 dos 8 estados** (i.e., $\{q_0\}, \{q_0, q_1\}$ e $\{q_0, q_f\}$) **são alcançáveis**. Portanto, a criação de um AFD a partir dos estados alcançáveis e suas transições resulta no autômato abaixo:



"Lazy evaluation"

- **Base**: O conjunto contendo somente o estado inicial de AFN (i.e., q_0) é acessível.

- **Passo de indução**: Supondo-se que foi determinado que o conjunto de estados S é acessível. Então, para cada símbolo de entrada a, calcula-se $\delta_D(S, a)$; sabe-se que os conjuntos resultantes também serão acessíveis.

“Lazy evaluation”

- **Base**: O conjunto contendo somente o estado inicial de AFN (i.e., q_0) é acessível.

- **Passo de indução**: Supondo-se que foi determinado que o conjunto de estados S é acessível. Então, para cada símbolo de entrada a, calcula-se $\delta_D(S, a)$; sabe-se que os conjuntos resultantes também serão acessíveis.

- Considerando o exemplo anterior:

δ	0	1

"Lazy evaluation"

- **Base**: O conjunto contendo somente o estado inicial de AFN (i.e., q_0) é acessível.

- **Passo de indução**: Supondo-se que foi determinado que o conjunto de estados S é acessível. Então, para cada símbolo de entrada a, calcula-se $\delta_D(S, a)$; sabe-se que os conjuntos resultantes também serão acessíveis.

➡ Considerando o exemplo anterior:

δ	0	1
$\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$

"Lazy evaluation"

- **Base**: O conjunto contendo somente o estado inicial de AFN (i.e., q_0) é acessível.

- **Passo de indução**: Supondo-se que foi determinado que o conjunto de estados S é acessível. Então, para cada símbolo de entrada a, calcula-se $\delta_D(S, a)$; sabe-se que os conjuntos resultantes também serão acessíveis.

➡ Considerando o exemplo anterior:

δ	0	1
$\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_0, q1\}$	$\{q_0, q_1\}$	$\{q_0, q_f\}$

"Lazy evaluation"

- **Base**: O conjunto contendo somente o estado inicial de AFN (i.e., q_0) é acessível.

- **Passo de indução**: Supondo-se que foi determinado que o conjunto de estados S é acessível. Então, para cada símbolo de entrada a, calcula-se $\delta_D(S, a)$; sabe-se que os conjuntos resultantes também serão acessíveis.

➡ Considerando o exemplo anterior:

δ	0	1
$\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_0, q1\}$	$\{q_0, q_1\}$	$\{q_0, q_f\}$
$*\{q_0, q_f\}$	$\{q_0, q_1\}$	$\{q_0\}$

Exercício

Exercício ①: Converta para AFD o seguinte AFN:

δ	0	1
$\rightarrow q_0$	$\{q_{f_1}, q_{f_2}\}$	$\{q_{f_1}\}$
$*q_{f_1}$	$\{q_1\}$	$\{q_{f_1}, q_1\}$
q_1	$\{q_{f_2}\}$	$\{q_0\}$
$*q_{f_2}$	$\emptyset$	$\{q_0\}$

1 Equivalência entre AFDs e AFNs

2 Demonstrando a equivalência entre AFDs e AFNs
- Prova por construção
- Prova por indução

3 Considerações finais

Prova (7)
Por indução no tamanho da palavra

É preciso mostrar formalmente que a abordagem de construção de subconjuntos funciona.

Teorema → *Equivalência entre AFD e AFN*

Se $\mathcal{M}_D = (\Sigma, Q_D, \delta_D, \{q_0\}, F_D)$ é um AFD criado a partir de $\mathcal{M}_N = (\Sigma, Q_N, \delta_N, q_0, F_N)$ por meio da construção de subconjuntos, então $L(\mathcal{M}_D) = L(\mathcal{M}_N)$.

A demonstração é por indução no tamanho da palavra. Deve-se demonstrar que (suponha w uma palavra qualquer de Σ^*):

$$\delta_D^*(\{q_0\}, w) = \delta_N^*(q_0, w)$$

Prova (8)

Por indução no tamanho da palavra

- **Base**: Seja w tal que $|w| = 0$. Portanto, $w = \varepsilon$. De acordo com as bases das definições das funções programas estendidas para AFDs e AFNs:

$$\delta^*_D(\{q_0\}, \varepsilon) = \{q_0\} \text{ se e somente se } \delta^*_N(q_0, \varepsilon) = \{q_0\}$$

o que é verdadeiro por definição.[2]

[2] Veja definição das funções programa estendidas.

Prova (8)

Por indução no tamanho da palavra

- **Base**: Seja w tal que $|w| = 0$. Portanto, $w = \varepsilon$. De acordo com as bases das definições das funções programas estendidas para AFDs e AFNs:

$$\delta^*_D(\{q_0\}, \varepsilon) = \{q_0\} \text{ se e somente se } \delta^*_N(q_0, \varepsilon) = \{q_0\}$$

o que é verdadeiro por definição.[2]

- **Hipótese de indução**: Seja w tal que $|w| = n$ e $n \geqslant 1$. Suponha que o seguinte seja verdadeiro:

$$\delta^*_D(\{q_0\}, w) = \{p_1, p_2, \dots, p_k\}$$

se e somente se

$$\delta^*_N(q_0, w) = \{p_1, p_2, \dots, p_k\}$$

[2] Veja definição das funções programa estendidas.

Prova (9)

Por indução no tamanho da palavra

- **Passo de indução**: Seja $w = xa$ (onde a é o último símbolo de w), tal que $|xa| = n + 1$ e $n \geqslant 1$.

A parte indutiva da definição da função programa estendida para AFN é:

$$\delta_N^*(q_0, w) = \bigcup_{i=1}^{k} \delta_N(p_i, a) \qquad (1)$$

A construção de subconjuntos:

$$\delta_D^*(\{p_1, p_2, \ldots, p_k\}, a) = \bigcup_{i=1}^{k} \delta_N(p_i, a) \qquad (2)$$

Prova (10)

Por indução no tamanho da palavra

Usando (2) e o fato que $\delta_D^*(\{q_0\}, x) = \{p_1, p_2, \ldots, p_k\}$ (veja a parte indutiva da definição da função programa estendida para AFDs):

$$\delta_D^*(\{q_0\}, w) = \delta_D(\delta_D^*(\{q_0\}, x), a) = \delta_D(\{p_1, p_2, \ldots, p_k\}, a) = \bigcup_{i=1}^{k} \delta_N(p_i, a) \tag{3}$$

Portanto, equações (1) e (3) demonstram que $\delta_D^*(\{q_0\}, w) = \delta_N^*(q_0, w)$. É possível afirmar que $\mathcal{M}_D$ e $\mathcal{M}_N$ aceitam w se e somente se $\delta_D^*(\{q_0\}, w)$ e $\delta_N^*(q_0, w)$ contêm um estado em F_N. Portanto, $L(\mathcal{M}_D) = L(\mathcal{M}_N)$. ❑

1. Equivalência entre AFDs e AFNs

2. Demonstrando a equivalência entre AFDs e AFNs
   - Prova por construção
   - Prova por indução

3. Considerações finais

Considerações finais. . .

Na aula de hoje nós vimos:

- Equivalência entre AFDs e AFNs;
  - Abordagem para construir um AFD a partir de um AFN.[3]
  - Prova (por indução).

Na **próxima aula**: autômatos finitos com movimentos vazios.

[3] A abordagem de construção de subconjuntos.

Referências

Hopcroft, John E., Rajeev Motwani, & Jeffrey D. Ullman (2006). *Introduction to Automata Theory, Languages, and Computation*. 3rd ed. Pearson, p. 750.

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!** ☺