

3.1 FA와 같은 Regular expression을 구하는 연립방정식을 이용한 방법

(개요) 연립방정식을 푼다.

DFA가 받아들이는 언어와 같은 regular expression을 구하는 이론적이고 어려운(따라서 좀 느린) 방법을 앞에서 recursion을 이용하여 구해보았다. 좀 더 분석적이고 간명한 방법을 생각해 보자.

확장된 FA¹⁾를 생각하자, 상태 q 에서 문자열 $x_1, x_2, \dots, x_n (1 \leq \forall i \leq n: x_i \in \Sigma^*)$ 을 보고 상태 p_1, p_2, \dots, p_n 으로 상태변환을 한다면 이것은

$$q = x_1p_1 + x_2p_2 + \dots + x_np_n$$

으로 표시할 수 있다. 상태 q 가 끝나는 상태 F 에 속한다면 $q = \epsilon$ 이 추가(+)²⁾된다.

$$\begin{aligned} q &= x_1p_1 + x_2p_2 + \dots + x_np_n + \epsilon && \text{if } q \in F, \\ q &= x_1p_1 + x_2p_2 + \dots + x_np_n && \text{otherwise}(q \notin F). \end{aligned}$$

만일 상태가 n 개 있다면, 상태에 관한 등식 n 개를 가질 수 있다. 상태를 미지수로 보고 상태의 1차 상수인 x_i 와 y_i 을 기지수로 본다면 이는 미지수 n 개에 대한 n 차 연립방정식으로 볼 수 있다.

$$\begin{aligned} 1 \leq \forall i \leq n: \quad q_i &= x_1q_1 + x_2q_2 + \dots + x_nq_n + y_i \\ &\text{if } q_i \in F \rightarrow y_i = \epsilon \mid q_i \notin F \rightarrow y_i = \emptyset \text{ fi} \end{aligned}$$

이 n 차 연립방정식을 초기상태에 관하여 풀면 우리는 오토마타가 받아들이는 언어를 표현하는 regular expression을 구할 수 있다²⁾.

연립방정식을 푸는 방법을 생각해 보자. 연립방정식을 푸는 기본적인 해법은 미지수(변수; 문자)를 차례로 제거하는 방법³⁾이다. 그런데 등식의 좌변에 있는 변수가 우변에도 나타났을 때⁴⁾ 우변에 있는 변수를 좌변으로 이항한 후 나누어서 해결한다.

$$q_i = x_iq_i + x_{i+1}q_{i+1} + \dots + x_nq_n + y_i \tag{1}$$

$$q_i = \frac{1}{1-x_i}(x_{i+1}q_{i+1} + \dots + x_nq_n + y_i) \tag{2}$$

그러나 regular repression은 빼기(이항)와 나누기 연산은 정의되어 있지 않으므로 (2)의 해법은 답이 아니다. 식 (1)을 아래 (3)과 같이 간단히 써보자.

$$A = \alpha A + \beta \tag{3}$$

1) $\delta: Q \times \Sigma^* \rightarrow 2^Q$

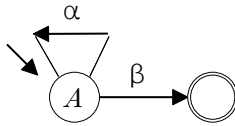
2) 최종상태 F 는 ϵ 으로 바뀌었음에 주의하라.

3) 대수학에서 Gauss-Jordan elimination을 배웠을 것이다.

4) Gauss-Jordan에서 non-zero diagonal element.

5) q_i 보다 작은 변수들(q_1, \dots, q_{i-1})은 Gauss-Jordan에 의하여 이미 제거되었다.

(3)식을 오토마타로 표현하여 보면,



이때 regular expression (3)의 해는

$$A = \alpha^* \beta \tag{4}$$

이다. 아래에 두 번 틀려서 맞은(?), 재미있는(?) 증명 방법을 생각해보자.

(틀린 증명)

$$\begin{aligned}
 A &= \alpha A + \beta \\
 \neq A &= \frac{1}{1-\alpha} \beta && \text{사칙연산의 경우} \\
 = A &= (1 + \alpha + \alpha^2 + \dots) \beta && \text{Taylor expansion} \\
 \neq A &= \alpha^* \beta && \text{*의 정의}
 \end{aligned}$$

위의 (틀린 증명)은 수학의 유연성을 보여주는 재미있는 예다.

(예 1) $(0+1)^*011$ 를 받아들이는 NFA \Rightarrow RE

$$\begin{aligned}
 A &= (0+1)A + 0B && \text{①} \\
 B &= 1C && \text{②} \\
 C &= 1D && \text{③} \\
 D &= \epsilon && \text{④}
 \end{aligned}$$

④를 ③에 대입 $C = 1$ ⑤

⑤를 ②에 대입 $B = 11$ ⑥

⑥을 ①에 대입 $A = (0+1)A + 011$ ⑦

⑦에 공식 (3), (4) 적용 $A = (0+1)^*011$ ⑧

(예 1)은 우리가 구성한 NFA가 맞는다는 증명이 되기도 한다.

(예 2) $(0+1)^*011$ 를 받아들이는 XFA \Rightarrow RE

$$\begin{aligned}
 A &= (0+1)A + 011D && \text{①} \\
 D &= \epsilon && \text{②}
 \end{aligned}$$

②를 ①에 대입 $A = (0+1)A + 011$ ③

③에 공식 적용 $A = (0+1)^*011$ ④

(예 2)는 확장한 XFA가 FA와 같다는 증명이 되기도 한다.

(예 3) $(0+1)^*011$ 를 받아들이는 DFA \Rightarrow RE₁

6) 식 (1)에서 q_i 는 A 로, x_i 는 α 로, $x_{i+1}q_{i+1} \dots y_i$ 는 β 로 각각 바꾸었다.

Regular expression의 연립방정식을 이용한 해법

$$A = 0B + 1A \quad \text{①}$$

$$B = 0B + 1C \quad \text{②}$$

$$C = 0B + 1D \quad \text{③}$$

$$D = 0B + 1A + \epsilon \quad \text{④}$$

$$\text{④를 ③에 대입} \quad C = 0B + 10B + 11A + 1 = (0 + 10)B + 11A + 1 \quad \text{⑤}$$

$$\text{⑤를 ②에 대입} \quad B = 0B + (10 + 110)B + 111A + 11 = (0 + 10 + 110)B + 111A + 11 \quad \text{⑥}$$

$$\text{⑥에 공식 적용} \quad B = (0 + 10 + 110)^*(111A + 11) \quad \text{⑦}$$

$$\begin{aligned} \text{⑦을 ①에 대입} \quad A &= 0(0 + 10 + 110)^*(111A + 11) + 1A \\ &= (0(0 + 10 + 110)^*111 + 1)A + 0(0 + 10 + 110)^*11 \\ &= (0(0 + 10 + 110)^*111 + 1)^*0(0 + 10 + 110)^*11 \quad \text{⑧} \end{aligned}$$

(예 4) $(0 + 1)^*011$ 를 받아들이는 DFA \Rightarrow RE₂

$$\text{①을 ④에 대입} \quad D = A + \epsilon \quad \text{⑤}$$

$$\text{⑤를 ③에 대입} \quad C = 0B + 1A + 1 \quad \text{⑥}$$

$$\text{①을 ⑥에 대입} \quad C = A + 1 \quad \text{⑦}$$

$$\text{⑦을 ②에 대입} \quad B = 0B + 1A + 11 \quad \text{⑧}$$

$$\text{①을 ⑧에 대입} \quad B = A + 11 \quad \text{⑨}$$

$$\text{⑨를 ①에 대입} \quad A = 0A + 011 + 1A = (0 + 1)A + 011 \quad \text{⑩}$$

$$\text{⑩에 공식 적용} \quad A = (0 + 1)^*011 \quad \text{⑪}$$

(예 3)과 (예 4)는 RE₁과 RE₂가 같다는 증명이다.