

3 オートマトンと正規表現

第 1 回・第 2 回ではグラフを扱った。今回からは、主題を変えて計算の数学的なモデルを扱う。計算機科学の中心的な問いの一つは「計算機に本質的にできること・できないことは何か」であり、その問いに答えるためには、まず「計算」や「機械」そのものを数学的に定式化しなければならない。今回はその最も基本的なモデルである有限オートマトン (finite automaton) を定義し、それが扱う文字列の集合 (言語) を調べる。さらに、同じ言語を記号列の式で表す正規表現 (regular expression) を導入する。両者が結局は同じ言語のクラスを定めること (クリーネの定理) は、今回末の補足で証明する (具体的な変換の手順と例は次回くわしく扱う)。

● 3-1 : 言語と文字列

まず、オートマトンへの入力となる「文字列」と、その集まりである「言語」を厳密に定める。

定義 3.1. 空でない有限集合 Σ を アルファベット (alphabet) という。以下、 Σ をアルファベットとする。

- (a) Σ の元を 文字 (letter) といい、文字を有限個並べた列を 文字列 (string) という。文字列全体のなす集合を Σ^* で表す。
- (b) 文字列 $w \in \Sigma^*$ に含まれる文字の個数を w の 長さ (length) といい、 $\ell(w)$ で表す。長さ 0 の文字列を 空列 (empty string) といい、特別な記号 ε で表す。空列でない文字列全体を $\Sigma^+ := \Sigma^* \setminus \{\varepsilon\}$ で表す。
- (c) Σ^* の部分集合を Σ 上の 言語 (language) という。

以下、アルファベットの文字は 0, 1, a, b のようにタイプライタ体で表す。二つの文字列 $w = a_1 \cdots a_m$, $w' = b_1 \cdots b_n$ に対し、それらを順に並べた文字列 $a_1 \cdots a_m b_1 \cdots b_n$ を w と w' の 連接 (concatenation) といい、単に ww' と書く。連接は結合法則 $(ww')w'' = w(w'w'')$ をみたし、空列 ε は $\varepsilon w = w\varepsilon = w$ をみたす単位元である。

例 3-1 アルファベット $\Sigma = \{0, 1\}$ をとる。このとき

$$\Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, \dots\}$$

であり、 $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$ である。文字列 $w = 0110$ は $\ell(w) = 4$ をみたし、 w と $w' = 10$ の連接は $ww' = 011010$ である。集合 $\{w \in \Sigma^* \mid w \text{ は } 1 \text{ で終わる}\}$ は Σ 上の言語の一例である。

● 3-2 : 決定性有限オートマトン

有限オートマトンは、有限個の「状態」をもち、入力文字列を左から 1 文字ずつ読みながら状態を変えていく機械である。文字列を読み終えたときの状態によって、その文字列を「受理」するか「拒否」するかが決まる。まずは、各状態と各入力文字に対して次の状態がただ一通りに定まる、決定性の場合を定義する。

定義 3.2. 二つの有限集合 K, Σ , 写像 $\delta: K \times \Sigma \rightarrow K$, および $q_0 \in K$ と部分集合 $F \subseteq K$ の 5 つ組 $M = (K, \Sigma, \delta, q_0, F)$ を 決定性有限オートマトン (deterministic finite automaton, DFA) という。ここで、

- K の元を 状態 (state) ,
- Σ を 入力アルファベット (input alphabet) ,
- $\delta: K \times \Sigma \rightarrow K$ を 遷移関数 (transition function) ,
- $q_0 \in K$ を 開始状態 (start state) ,
- F の元を 受理状態 (accepting state)

という。

決定性有限オートマトンは [状態遷移図](#) (state diagram) によって視覚的に表せる. 各状態を丸で表し, $\delta(q, a) = q'$ を q から q' への a を付した矢印で表す. 開始状態には外から矢印を付し, 受理状態は二重丸で表す. 遷移関数は表 (遷移表) でも表せる. 表では, 開始状態の行に $\rightarrow$ を, 受理状態の行に $*$ を付す.

例 3-2 $K = \{q_0, q_1\}$, $\Sigma = \{0, 1\}$, $F = \{q_1\}$ とし, 遷移関数 δ を次の表で与える.

δ		0	1
$\rightarrow q_0$	q_0	q_1	
$*q_1$	q_0	q_1	

この M に文字列 101 を入力すると, 状態は

$$q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_0 \xrightarrow{1} q_1$$

と動く. 読み終えた状態 q_1 は受理状態だから, M は 101 を受理する.

● 3-3 : オートマトンが認識する言語

「文字列を読み終えたときの状態」を厳密に述べるため, 1 文字に対する遷移 δ を, 文字列に対する遷移 $\hat{\delta}$ へ拡張する.

定義 3.3. 決定性有限オートマトン $M = (K, \Sigma, \delta, q_0, F)$ に対し, [拡張遷移関数](#) $\hat{\delta}: K \times \Sigma^* \rightarrow K$ を, $(q, w) \in K \times \Sigma^*$ について次のように帰納的に定める.

- (i) $w = \varepsilon$ のとき, $\hat{\delta}(q, \varepsilon) = q$.
- (ii) $w = xa$ ($x \in \Sigma^*$, $a \in \Sigma$) のとき, $\hat{\delta}(q, w) = \delta(\hat{\delta}(q, x), a)$.

特に $a \in \Sigma$ に対して $\hat{\delta}(q, a) = \delta(q, a)$ である.

定義 3.4. 決定性有限オートマトン $M = (K, \Sigma, \delta, q_0, F)$ が文字列 $w \in \Sigma^*$ を [受理する](#) (accept) とは, $\hat{\delta}(q_0, w) \in F$ となることをいう. 受理しないとき, M は w を [拒否する](#) (reject) という. M が受理する文字列全体

$$T(M) := \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F\}$$

を, M が [認識する言語](#) (language recognized by M) という.

例 3-3 例 3-2 の M について, 入力 101 に対する拡張遷移関数の値を順に計算すると

$$\hat{\delta}(q_0, 1) = q_1, \quad \hat{\delta}(q_0, 10) = \delta(q_1, 0) = q_0, \quad \hat{\delta}(q_0, 101) = \delta(q_0, 1) = q_1 \in F$$

となり, 101 は受理される. 一般に, この M は最後に読んだ文字が 1 のときちょうど状態 q_1 にいるから,

$$T(M) = \{w \in \Sigma^* \mid w \text{ は } 1 \text{ で終わる}\}$$

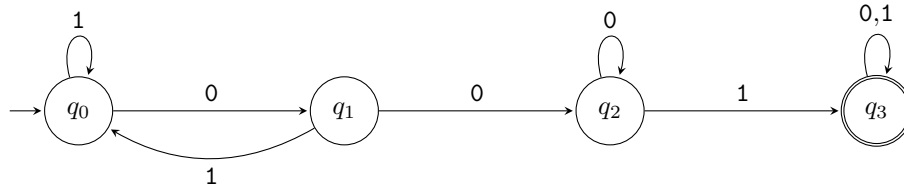
である (空列 ε は $q_0 \notin F$ より受理されない).

例 3-4 (オートマトンの設計) 逆に, 与えられた言語を認識するオートマトンを設計してみよう. $\Sigma = \{0, 1\}$ 上の言語

$$L = \{w \in \Sigma^* \mid w \text{ は } 001 \text{ を部分文字列として含む}\}$$

を考える. 「001 をどこまで読めたか」を状態にとればよい: まだ何も ($=q_0$), 0 まで ($=q_1$), 00 まで ($=q_2$), 001 を読み終えた ($=q_3$). 一度 001 を含めば以後はずっと含むので, q_3 は吸収状態かつ受理状態とする. 得られる $M = (K, \Sigma, \delta, q_0, \{q_3\})$ は次の通りで, $T(M) = L$ となる.

δ	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_0
q_2	q_2	q_3
$*q_3$	q_3	q_3



● 3-4 : 非決定性有限オートマトン

決定性有限オートマトンでは、各状態・各文字に対して次の状態がただ一つに定まる。これを緩めて、次の状態の候補が複数（または 0 個）あってもよいとしたものが非決定性有限オートマトンである。このとき遷移先は「状態の集合」となるので、遷移関数の終域を K のべき集合 2^K とする。

定義 3.5. 二つの有限集合 K, Σ , 写像 $\delta: K \times \Sigma \rightarrow 2^K$, および $q_0 \in K$ と $F \subseteq K$ の 5 つ組 $M = (K, \Sigma, \delta, q_0, F)$ を **非決定性有限オートマトン** (nondeterministic finite automaton, NFA) という。ここで 2^K は K の部分集合全体のなす集合である。

非決定性有限オートマトンに文字列を入力すると、各段階で複数の遷移先がありうるため、状態は「枝分かれ」しながら進む。入力を読み終えたとき、到達する状態の集合のうち 1 つでも受理状態があれば、その文字列を受理する。これを厳密に述べるため、拡張遷移関数を次のように定める。

定義 3.6. 非決定性有限オートマトン $M = (K, \Sigma, \delta, q_0, F)$ に対し、拡張遷移関数 $\hat{\delta}: K \times \Sigma^* \rightarrow 2^K$ を次のように帰納的に定める。

(i) $w = \varepsilon$ のとき、 $\hat{\delta}(q, \varepsilon) = \{q\}$.

(ii) $w = xa$ ($x \in \Sigma^*$, $a \in \Sigma$) で $\hat{\delta}(q, x) = \{r_1, \dots, r_k\}$ のとき、 $\hat{\delta}(q, w) = \bigcup_{i=1}^k \delta(r_i, a)$.

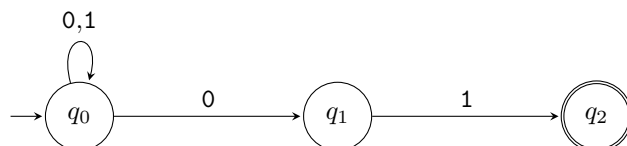
定義 3.7. 非決定性有限オートマトン $M = (K, \Sigma, \delta, q_0, F)$ が文字列 $w \in \Sigma^*$ を受理するとは、 $\hat{\delta}(q_0, w) \cap F \neq \emptyset$ となることをいう。 M が認識する言語を

$$T(M) := \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \cap F \neq \emptyset\}$$

で定める。

例 3-5 $K = \{q_0, q_1, q_2\}$, $\Sigma = \{0, 1\}$, $F = \{q_2\}$ とし、遷移関数 δ を次の表で与える ($\emptyset$ は遷移先がないことを表す)。

δ	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\emptyset$	$\{q_2\}$
$*q_2$	$\emptyset$	$\emptyset$



状態 q_0 は 0 を読むと q_0 にとどまる枝と q_1 へ進む枝に分かれる。 q_1 で 1 を読めば受理状態 q_2 に至る。ゆえにこの M は「末尾の 2 文字が 01 である」文字列をちょうど受理する： $T(M) = \{w \in \Sigma^* \mid w \text{ は } 01 \text{ で終わる}\}$.

たとえば入力 001 に対し

$$\widehat{\delta}(q_0, 0) = \{q_0, q_1\}, \quad \widehat{\delta}(q_0, 00) = \{q_0, q_1\}, \quad \widehat{\delta}(q_0, 001) = \{q_0, q_2\}$$

となり, $\{q_0, q_2\} \cap F = \{q_2\} \neq \emptyset$ だから 001 は受理される.

● 3-5 : 部分集合構成法

非決定性は記述を簡潔にするが, 「状態の集合」を一つの状態とみなせば, 非決定性有限オートマトンは決定性有限オートマトンに変換できる. これを **部分集合構成法** (subset construction) という.

構成. 非決定性有限オートマトン $M = (K, \Sigma, \delta, q_0, F)$ に対し, 決定性有限オートマトン $M^{\text{Det}} = (K', \Sigma, \delta', q'_0, F')$ を次で定める.

$$K' = 2^K, \quad q'_0 = \{q_0\}, \quad \delta'(S, a) = \bigcup_{r \in S} \delta(r, a) \quad (S \subseteq K, a \in \Sigma), \quad F' = \{S \subseteq K \mid S \cap F \neq \emptyset\}.$$

すなわち M^{Det} の状態は M の状態の集合であり, M が到達しうる状態の集合をひとまとめに追跡する.

定理 3.8. 任意の非決定性有限オートマトン M に対し, $T(M^{\text{Det}}) = T(M)$ が成り立つ.

証明. M と M^{Det} の拡張遷移関数をそれぞれ $\widehat{\delta}, \widehat{\delta}'$ と書く. まず, 任意の $w \in \Sigma^*$ について

$$\widehat{\delta}'(\{q_0\}, w) = \widehat{\delta}(q_0, w) \quad \dots\dots\dots \textcircled{1}$$

が成り立つことを, 長さ $\ell(w)$ に関する帰納法で示す. $\ell(w) = 0$, すなわち $w = \varepsilon$ のときは, 定義より両辺とも $\{q_0\}$ に等しい. 次に, ある $x \in \Sigma^*$ について $\textcircled{\text{Equation.1}}$ が成り立つと仮定し, $w = xa$ ($a \in \Sigma$) とする. このとき, δ' の定義と帰納法の仮定により

$$\widehat{\delta}'(\{q_0\}, xa) = \delta'(\widehat{\delta}'(\{q_0\}, x), a) = \delta'(\widehat{\delta}(q_0, x), a) = \bigcup_{r \in \widehat{\delta}(q_0, x)} \delta(r, a) = \widehat{\delta}(q_0, xa)$$

となり, 最後の等号は M の拡張遷移関数の定義による. ゆえに $\textcircled{\text{Equation.1}}$ が示された. これを用いると, 各 $w \in \Sigma^*$ について

$$w \in T(M^{\text{Det}}) \iff \widehat{\delta}'(\{q_0\}, w) \in F' \iff \widehat{\delta}'(\{q_0\}, w) \cap F \neq \emptyset \iff \widehat{\delta}(q_0, w) \cap F \neq \emptyset \iff w \in T(M)$$

が成り立つ (2 番目の同値は F' の定義, 3 番目は $\textcircled{\text{Equation.1}}$ による). ゆえに $T(M^{\text{Det}}) = T(M)$ である. $\square$

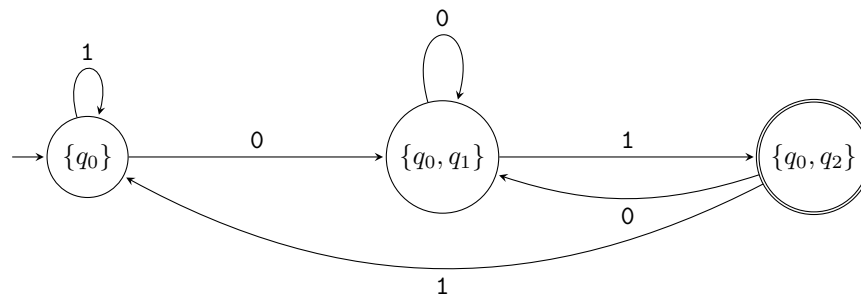
定理 3.9. 言語 L がある決定性有限オートマトンに認識されるための必要十分条件は, L がある非決定性有限オートマトンに認識されることである.

証明. (十分性) L を認識する非決定性有限オートマトン M があれば, 定理 3.8 より決定性有限オートマトン M^{Det} が $L = T(M) = T(M^{\text{Det}})$ を認識する. (必要性) 逆に L を認識する決定性有限オートマトン $M = (K, \Sigma, \delta, q_0, F)$ があれば, $\delta_N(q, a) := \{\delta(q, a)\}$ により非決定性有限オートマトン $N = (K, \Sigma, \delta_N, q_0, F)$ を定めると, $\ell(w)$ に関する容易な帰納法で $\widehat{\delta}_N(q_0, w) = \{\widehat{\delta}(q_0, w)\}$ が示され, $T(N) = T(M) = L$ となる. $\square$

例 3-6 例 3-5 の非決定性有限オートマトンに部分集合構成法を適用する. 開始状態 $\{q_0\}$ から到達できる状態 (部分集合) を順に求めると

$$\begin{aligned} \{q_0\} &\xrightarrow{0} \{q_0, q_1\}, & \{q_0\} &\xrightarrow{1} \{q_0\}, & \{q_0, q_1\} &\xrightarrow{0} \{q_0, q_1\}, & \{q_0, q_1\} &\xrightarrow{1} \{q_0, q_2\}, \\ & & \{q_0, q_2\} &\xrightarrow{0} \{q_0, q_1\}, & \{q_0, q_2\} &\xrightarrow{1} \{q_0\} \end{aligned}$$

となり, 到達可能な状態は $\{q_0\}, \{q_0, q_1\}, \{q_0, q_2\}$ の 3 つである. 受理状態は q_2 を含む $\{q_0, q_2\}$ のみ. 得られる決定性有限オートマトン (到達不能な状態は除いた) は次の通りで, 確かに「01 で終わる」文字列を認識する.



● 3-6 : 正規表現

オートマトンが「機械」として言語を定めるのに対し、正規表現は言語を「式」として簡潔に書き表す方法である。準備として、言語に対する 3 つの演算を定める。

定義 3.10. アルファベット Σ 上の言語 $A, B \subseteq \Sigma^*$ に対し、次の演算を **正規演算** (regular operations) という。

- **和** (union) : $A \cup B = \{w \mid w \in A \text{ または } w \in B\}$.
- **接続** (concatenation) : $A \circ B = \{ww' \mid w \in A, w' \in B\}$.
- **スター** (star) : $A^* = \{w_1w_2 \cdots w_k \mid k \geq 0, w_i \in A (1 \leq i \leq k)\}$. ただし $k = 0$ の場合は空列 ε を表すものとし、つねに $\varepsilon \in A^*$ である。

例 3-7 $\Sigma = \{0, 1\}$ とし, $A = \{0, 11\}$, $B = \{00, 01\}$ とする. このとき

$$A \cup B = \{0, 11, 00, 01\}, \quad A \circ B = \{000, 001, 1100, 1101\},$$

$$A^* = \{\varepsilon, 0, 11, 00, 011, 110, 1111, \dots\} \text{ (0 と 11 を任意個並べた文字列全体)}$$

である。

これらの演算を記号の上で組み立てたものが正規表現である。構文（式の作り方）と意味（表す言語）を同時に、帰納的に定義する。

定義 3.11. アルファベット Σ 上の **正規表現** (regular expression) E と, E が **表す言語** $\mathcal{L}(E) \subseteq \Sigma^*$ を, 次のように帰納的に定める。

- (I) (基礎) $\emptyset$ と ε は正規表現であり, $\mathcal{L}(\emptyset) = \emptyset$, $\mathcal{L}(\varepsilon) = \{\varepsilon\}$. また各 $a \in \Sigma$ は正規表現であり, $\mathcal{L}(a) = \{a\}$.
- (II) (帰納) E, F が正規表現ならば, 次も正規表現であり, 表す言語は右の通りである。

$$\mathcal{L}(E + F) = \mathcal{L}(E) \cup \mathcal{L}(F), \quad \mathcal{L}(EF) = \mathcal{L}(E) \circ \mathcal{L}(F), \quad \mathcal{L}(E^*) = \mathcal{L}(E)^*, \quad \mathcal{L}((E)) = \mathcal{L}(E).$$

以上の手続きで得られるものだけを正規表現とする。演算子の結合の強さは, 強い順に スター *, 接続 (併置), 和 + とし, 括弧で優先順位を指定できる。

言語の演算と正規表現の記号は区別している: 和は言語では $\cup$, 正規表現では + と書き, 接続は言語では $\circ$, 正規表現では記号を書かずに併置する。

例 3-8 $\Sigma = \{0, 1\}$ 上の正規表現の例を挙げる。

- (1) $(0 + 1)^*$ は Σ^* 全体 (すべての文字列) を表す。
- (2) 0^*10^* は「1 をちょうど 1 つ含む」文字列全体を表す。
- (3) $(0 + 1)^*1$ は「1 で終わる」文字列全体, すなわち例 3-2 の $T(M)$ を表す。

(4) $(0+1)^*01$ は「01 で終わる」文字列全体, すなわち例 3-5 の $T(M)$ を表す.

(5) $(0+1)^*001(0+1)^*$ は「001 を含む」文字列全体, すなわち例 3-4 の L を表す.

これらの例では, 同じ言語が「オートマトンが認識する言語」としても「正規表現が表す言語」としても現れている. これは偶然ではない: 実は, ある言語が有限オートマトンに認識されることと, ある正規表現で表せることは同値である. このとき, その言語を **正規言語** (regular language) という. この同値性は **クリーネの定理** (Kleene's theorem) と呼ばれ, 「正規表現から有限オートマトンへ」「有限オートマトンから正規表現へ」という両方向の変換によって証明される. その証明を本回末の補足 (● 3-A) に与える (変換の具体的な手順と例は次回くわしく扱う).

● 3-A : クリーネの定理の証明 (補足)

第 3-6 節で名を挙げた次の定理を証明する.

定理 3.12 (クリーネの定理). 言語 $L \subseteq \Sigma^*$ について, 次の 2 つは同値である.

- (1) L はある有限オートマトンに認識される (すなわち L は正規言語である).
- (2) L はある正規表現で表せる.

定理 3.9 より「決定性有限オートマトンに認識される」と「非決定性有限オートマトンに認識される」ことは同値だから, 以下では有限オートマトンとして非決定性のものを用いてよい. 構成を簡潔にするため, 文字を読まずに状態を移れる遷移 (ε 遷移) を許したものを補助に使う. 遷移関数が $\delta: K \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^K$ である 5 つ組を **ε -NFA** という. 状態 q から ε 遷移だけをたどって到達できる状態全体 (q を含む) を q の **ε 閉包** といい, $E(q)$ で表す (集合 S には $E(S) = \bigcup_{q \in S} E(q)$). ε -NFA は, 各文字の読み取りの前後に ε 遷移を自由にはさんで w を読み, 受理状態に到達できるとき w を受理するものとする.

命題 3.13. 任意の ε -NFA に対し, それと同じ言語を認識する (ε 遷移をもたない) 非決定性有限オートマトンが存在する.

証明. ε -NFA $M = (K, \Sigma, \delta, q_0, F)$ から, 非決定性有限オートマトン $N = (K, \Sigma, \delta_N, q_0, F_N)$ を

$$\delta_N(q, a) = E(\delta(E(q), a)) \quad (a \in \Sigma), \quad F_N = \{q \in K \mid E(q) \cap F \neq \emptyset\}$$

で定める (ここで $\delta(S, a) = \bigcup_{r \in S} \delta(r, a)$). 文字列 w に対し, M において q_0 から ε 遷移を自由にはさみつつ w を読み終えて到達できる状態の集合を $R(w)$ とおく. 定義より $R(\varepsilon) = E(q_0)$, $R(xa) = E(\delta(R(x), a))$ であり, M が w を受理することと $R(w) \cap F \neq \emptyset$ は同値である. また各 $R(w)$ は ε 閉じている ($E(R(w)) = R(w)$).

まず, 空でない文字列 w について $\widehat{\delta}_N(q_0, w) = R(w)$ が成り立つことを $\ell(w)$ に関する帰納法で示す (空列では $\widehat{\delta}_N(q_0, \varepsilon) = \{q_0\}$ であり, 一般に $R(\varepsilon) = E(q_0)$ とは異なるので別に扱う). $\ell(w) = 1$, すなわち $w = a$ ($a \in \Sigma$) のときは

$$\widehat{\delta}_N(q_0, a) = \delta_N(q_0, a) = E(\delta(E(q_0), a)) = E(\delta(R(\varepsilon), a)) = R(a).$$

空でない x について主張を仮定し $w = xa$ とすると, 写像 $S \mapsto E(\delta(S, a))$ が和集合を保つことと $\bigcup_{r \in R(x)} E(r) = R(x)$ ($R(x)$ の ε 閉性) から

$$\widehat{\delta}_N(q_0, xa) = \bigcup_{r \in R(x)} \delta_N(r, a) = \bigcup_{r \in R(x)} E(\delta(E(r), a)) = E(\delta(R(x), a)) = R(xa)$$

となる.

次に受理を比べる. $R(w)$ が ε 閉じていることから, $q \in R(w)$ ならば $E(q) \subseteq R(w)$ であり, ゆえに「ある $q \in R(w)$ について $E(q) \cap F \neq \emptyset$ 」と「 $R(w) \cap F \neq \emptyset$ 」は同値である. したがって空でない w では, 上で示

した等式と F_N の定義により

$$w \in T(N) \iff \widehat{\delta}_N(q_0, w) \cap F_N \neq \emptyset \iff R(w) \cap F \neq \emptyset \iff w \in T(M)$$

が成り立つ. また空列については, $\widehat{\delta}_N(q_0, \varepsilon) = \{q_0\}$ より

$$\varepsilon \in T(N) \iff q_0 \in F_N \iff E(q_0) \cap F \neq \emptyset \iff R(\varepsilon) \cap F \neq \emptyset \iff \varepsilon \in T(M)$$

が成り立つ. 以上より $T(N) = T(M)$ である. $\square$

この命題より, 「 ε -NFA に認識される」ことも条件 (1) と同値である.

定理 3.12 の証明. (2) $\Rightarrow$ (1) (正規表現から有限オートマトンへ). 正規表現 E の構成に関する帰納法で, $\mathcal{L}(E)$ を認識する ε -NFA $M[E]$ が作れることを示す. ただし $M[E]$ は, 開始状態と受理状態をちょうど 1 つずつもち, 開始状態に入る辺・受理状態から出る辺をもたない形に保つ.

(基礎) $\emptyset, \varepsilon, a$ に対しては, 開始状態 s と受理状態 f をとり, それぞれ「辺をもたない」「 $s \xrightarrow{\varepsilon} f$ 」「 $s \xrightarrow{a} f$ 」とすればよい.

(帰納) $M[E_1], M[E_2]$ が得られているとする.

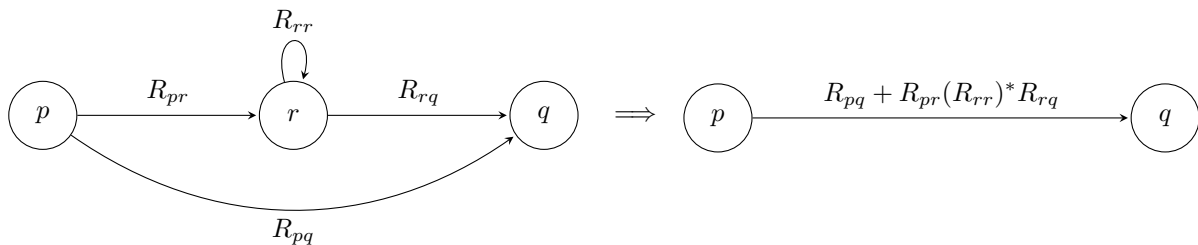
- 和 $E_1 + E_2$: 新たな開始状態 s と受理状態 f を加え, s から $M[E_1], M[E_2]$ の各開始状態へ ε 遷移を, $M[E_1], M[E_2]$ の各受理状態から f へ ε 遷移を引く.
- 接続 $E_1 E_2$: $M[E_1]$ の受理状態から $M[E_2]$ の開始状態へ ε 遷移を引き, $M[E_1]$ の開始状態を全体の開始状態, $M[E_2]$ の受理状態を全体の受理状態とする.
- スター E_1^* : 新たな開始状態 s と受理状態 f を加え, $s \xrightarrow{\varepsilon} f$ と, s から $M[E_1]$ の開始状態へ, $M[E_1]$ の受理状態から $M[E_1]$ の開始状態へ, $M[E_1]$ の受理状態から f へ, それぞれ ε 遷移を引く.

ε 遷移のたどり方を追えば, これらが認識する言語はそれぞれ $\mathcal{L}(E_1) \cup \mathcal{L}(E_2)$, $\mathcal{L}(E_1) \circ \mathcal{L}(E_2)$, $\mathcal{L}(E_1)^*$ に等しい. ゆえに任意の正規表現 E について $\mathcal{L}(E)$ は ε -NFA に認識され, 命題 3.13 より正規言語である.

(1) $\Rightarrow$ (2) (有限オートマトンから正規表現へ). L を認識する有限オートマトン $M = (K, \Sigma, \delta, q_0, F)$ をとる. 辺のラベルに正規表現を許した図を考える (もとの遷移 $q' \in \delta(p, a)$ は辺 $p \xrightarrow{a} q'$ とみなし, 同じ 2 状態間の複数の文字は和 $+$ でまとめる). まず, 新たな開始状態 s_0 と受理状態 f_0 を加え, $s_0 \xrightarrow{\varepsilon} q_0$ と, 各受理状態 $q \in F$ について $q \xrightarrow{\varepsilon} f_0$ を引く. 次に, s_0, f_0 以外の状態 r を一つ選び, r を経由しうる各 2 状態 p, q ($p, q \neq r$) の組について, $p \rightarrow q$ の辺のラベルを

$$R_{pq} + R_{pr}(R_{rr})^* R_{rq}$$

に書き換え, r と r に接続する辺を消す (R_{pq} は $p \rightarrow q$ の既存ラベル, R_{rr} は r の自己ループのラベルで, いずれも無い箇所は $\emptyset$ とみなす. なお $\emptyset^* = \varepsilon$ である).



この書き換えで, 図がつづる文字列の全体 (s_0 から f_0 への経路に沿ってラベルの正規表現を接続して得られる言語) は変わらない. 中間状態をすべて消すと $s_0 \xrightarrow{R} f_0$ の 1 本だけが残る, このラベル R が $\mathcal{L}(R) = L$ をみたす正規表現である. ゆえに L は正規表現で表せる.

以上で (1) と (2) の同値が示され, クリーネの定理が証明された. $\square$