9　形式言語論 (3) – 正規表現から有限オートマトンへ –

正規表現は, 有限オートマトンとは異なる考え方で言語を正確に表現するための記法である. この記法は, 一種のプログラミング言語としての側面をもち, 重要な応用例として「テキスト検索」や「コンパイラ」がある. ここでは, 正規表現が表現する言語は有限オートマトンによって認識できることを示す.

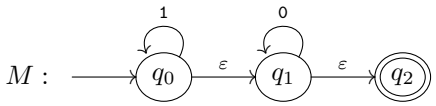
• 9-1 : ε-規則を許した非決定性有限オートマトン

まず, 非決定性有限オートマトンに「ε-規則」を許してその表現力を高めよう. 「ε-規則」は, 外部からの入力を受けなくとも自動的に状態を変化する動作を指す.

ε-規則を許す非決定性有限オートマトン とは, 2 つの有限集合 K, Σ と写像

$$\delta : K \times (\Sigma \cup \{\varepsilon\}) \longrightarrow 2^K$$

および $q_0 \in K$ と $F \subset K$ の 5 つ組 $M = (K, \Sigma, \delta, q_0, F)$ である. 通常と違う点は, 遷移関数の定義域として, 「状態とアルファベットの組」だけでなく「状態と空文 ε との組」に対しても値を定義していることである.



この非決定性オートマトンは, まず 1 を任意の個数読み込み, ε によって状態 q_1 に遷移する. その後, 0 を任意の個数読み込み, 再び ε によって受理状態へ遷移する.

ε-規則を許すことで, その表現力が高まったわけであるが, 実は認識できる言語のクラスは広がらない. つまり, 次の定理が成り立つ[*1].

定理 9.1. 言語 L がある ε-規則を許す非決定性有限オートマトン M で認識されるための必要十分条件は, L がある ε-規則を許さない非決定性有限オートマトン M' で認識されることである.

• 9-2 : 言語の正規演算

定義 9.2. 言語 A, B に対して, 以下で定義される 3 つの演算を **正規演算** という.

- **和集合演算** : $A \cup B := \{w \mid (w \in A) \vee (w \in B)\}$
- **連接演算** : $A \circ B := \{ww' \mid (w \in A) \wedge (w' \in B)\}$
  ただし, A もしくは B が空集合のときは, $A \circ B = \varnothing$ と定める.
- **スター演算** : $A^* := \{w_1 w_2 \cdots w_k \mid (k \in \mathbb{N} \cup \{0\}) \wedge (\forall i \in \{1, 2, \ldots, k\},\ w_i \in A)\}$
  ただし, $k = 0$ のときは ε と定める.

任意の言語 A, B, C に対して, 和集合演算と連接演算は結合法則

$$(A \cup B) \cup C = A \cup (B \cup C), \quad (A \circ B) \circ C = A \circ (B \circ C)$$

が成り立つ. そこで, 任意の $k \in \mathbb{N}$ に対して,

$$A^{\circ k} := \overbrace{A \circ A \circ \cdots \circ A}^{k}$$

と定義することができる. これを用いると, 言語 A のスター演算 A^* は, 以下のように帰納的に定めるのが正確である.

$$A^{\circ 0} := \{\varepsilon\}, \quad A^* := \bigcup_{k=0}^{\infty} A^{\circ i}$$

[*1] 証明のアイデアは **9-A** 節を参照せよ.

例 9-1　$A := \{0, 11\}$, $B := \{00, 01, 11\}$ とする．このとき，

$$A \cup B = \{0, 00, 01, 11\}, \quad A \circ B = \{000, 001, 011, 1100, 1101, 1111\}$$

となる．また，

$$A^{\circ 0} = \{\varepsilon\}, \quad A^{\circ 1} = \{0, 11\}, \qquad A^{\circ 2} = \{00, 011, 110, 1111\},$$
$$A^{\circ 3} = \{000, 0011, 0110, 01111, 1100, 11110, 11011, 111111\}, \ldots$$

となり，A^* はこれら和集合演算をとったものである．つまり，A^* に属する文字列は，0 と 11 をいくつか用いて好きな順序で並べた文字列である．

• 9-3 ： 正規表現の定義

定義 9.3. Σ をアルファベットとする．Σ 上の**正規表現** E と E が表す言語 $\mathcal{L}(E)$ を以下で帰納的に定義する．

（基礎） (I-i) ε および $\varnothing$ は正規表現である．このとき，$\mathcal{L}(\varepsilon) := \{\varepsilon\}$, $\mathcal{L}(\varnothing) := \varnothing$ である．
(I-ii) 任意の $a \in \Sigma$ は正規表現である．このとき，$\mathcal{L}(a) := \{a\}$ である．
（帰納） (II-i) E と F が正規表現のとき，**和演算** $E + F$ も正規表現であり，$\mathcal{L}(E + F) := \mathcal{L}(E) \cup \mathcal{L}(F)$ である．
(II-ii) E と F が正規表現のとき，**連接演算** EF も正規表現であり，$\mathcal{L}(EF) := \mathcal{L}(E) \circ \mathcal{L}(F)$ である．
(II-iii) E が正規表現のとき，**スター演算** E^* も正規表現であり，$\mathcal{L}(E^*) := \mathcal{L}(E)^*$ である．
(II-iv) E が正規表現のとき，(E) も正規表現であり，$\mathcal{L}((E)) := \mathcal{L}(E)$ である．
（限定） 以上の手続きで得られるものだけ正規表現 E であり，$\mathcal{L}(E)$ は (I), (II) を満たす最小の集合である．

通常の数の場合，例えば $5 + 3 \times 4$ とある場合は積 3×4 を先に実行してその後に和 $5 + 12$ を計算する．正規演算に対しても，指定がなければその順序の優先順序が定まっている．

- **スター演算は優先順位が最も高い．** すなわち，スター演算が現れた場合は，その左にある最短の正規表現に適用される．
- **スター演算の次に，連接演算の優先順位が高い．**　まず，スター演算をすべて実行した後に，隣り合っている正則表現の間の連接演算を実行する．
- **最後に，和演算を適用する．**

連接演算と和演算は結合法則を満たすので，連続して同じ演算を適用する場合は，その演算順序は気にしなくても良い．また，(　) でくくることは，例えば $(3 + 4) \times 5$ において $3 + 4$ を優先して計算するように指定する方法であるが，同様にして (　) でくくることで，その演算を何よりも優先して実行するように指定できる．

例 9-2　以下の例において，$\Sigma = \{0, 1\}$ とする．

(1) $E = (0 + 1)^*$ は正規表現であり，$\mathcal{L}(E) = (\mathcal{L}(0) \cup \mathcal{L}(1))^* = (\{0\} \cup \{1\})^* = \Sigma^*$.
(2) $E = 0^*10^*$ は正規表現であり，

$$\mathcal{L}(E) = \{0\}^* \circ \{1\} \circ \{0\}^* = \{w \in \Sigma^* \mid w \text{ は } 1 \text{ をちょうど } 1 \text{ つ含む }\}.$$

(3) $E = (0 + 1)^*1(0 + 1)^*$ は正規表現であり，

$$\mathcal{L}(E) = \Sigma^* \circ \{1\} \circ \Sigma^* = \{w \in \Sigma^* \mid w \text{ は } 1 \text{ を少なくとも } 1 \text{ つ含む }\}.$$

(4) $E = ((0 + 1)(0 + 1))^*$ は正規表現であり，

$$\mathcal{L}(E) = (\Sigma \circ \Sigma)^* = \{w \in \Sigma^* \mid \ell(w) \text{ は } 0 \text{ 以上の偶数 }\}.$$

(5) $E = \mathtt{0}(\mathtt{0}+\mathtt{1})^*\mathtt{0} + \mathtt{1}(\mathtt{0}+\mathtt{1})^*\mathtt{1} + \mathtt{0} + \mathtt{1}$ は正規表現であり,

$$\begin{aligned}\mathcal{L}(E) &= \{\mathtt{0}\} \circ \Sigma^* \circ \{\mathtt{0}\} \cup \{\mathtt{1}\} \circ \Sigma^* \circ \{\mathtt{1}\} \cup \{\mathtt{0}\} \cup \{\mathtt{1}\} \\ &= \{w \in \Sigma^* \mid w \text{ は先頭と末尾の文字が同じである長さ } 1 \text{ 以上の文字列 }\}.\end{aligned}$$

(6) $E = (\mathtt{0}+\varepsilon)\mathtt{1}^*$ は正規表現であり,

$$\mathcal{L}(E) = (\{\mathtt{0}\} \cup \{\varepsilon\}) \circ \{\mathtt{1}\}^* = \{\mathtt{0},\varepsilon\} \circ \{\mathtt{1}\}^* = \{\mathtt{0}\overbrace{\mathtt{1}\cdots\mathtt{1}}^{k},\ \overbrace{\mathtt{1}\cdots\mathtt{1}}^{k} \mid k \in \mathbb{N} \cup \{0\}\}$$

(7) $E = \mathtt{1}^*\varnothing$ は正規表現であり, $\mathcal{L}(E) = \{\mathtt{1}\}^* \circ \varnothing = \varnothing$.

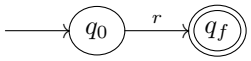
(8) $E = \varnothing^*$ は正規表現であり, $\mathcal{L}(E) = \{\varepsilon\}$.

- **9-4 : 正規表現から有限オートマトンへの変換**

正規表現と有限オートマトンは, 全く別の文脈で登場した. しかし, 後にみるようにこれらの表現力は等価である. つまり, 正規表現が表す言語は, ある有限オートマトンによって認識する言語と一致して, 逆に有限オートマトンによって認識する言語は, ある正規表現によって表すことができる. ここでは, 与えられた正規表現が表す言語と同じものを認識する非決定性有限オートマトンを構成するレシピを与える.

（手順 1） まず, アルファベット Σ 上の正規表現 r が与えられたとき, 以下の状態遷移図をもつ非決定性有限オートマトン M_0 を用意する.

M_0 :



（手順 2） r の表示によって, この状態遷移図 M_0 を更新してゆくように非決定性有限オートマトンを順次構成してゆく. ここで, Q は新たに追加される状態である.

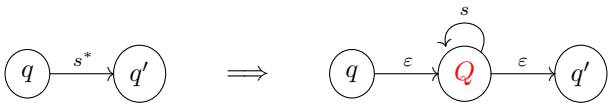
(2-1) $\underline{(r = \varepsilon) \vee (r = \varnothing) \vee (r \in \Sigma)}$

このときは, M_0 のまま, 更新せず **（手順 3）** へ.

(2-2) r が以下の形を含む場合は, 全ての矢印が ε, $\varnothing$ もしくは Σ に属するアルファベットによってラベル付けされているような状態遷移図を得るまで, 以下の更新を行う.

(2-2-1) $\underline{s^*}$

正規表現 r がスター演算 s^* の形を含むとき, 状態遷移図を次のように更新する.

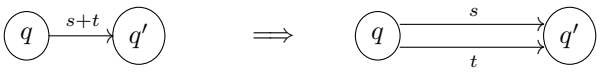


(2-2-2) $\underline{st}$

正規表現 r が連接演算 st の形を含むとき, 状態遷移図を次のように更新する.

(2-2-3) $\underline{s + t}$

正規表現 r が和演算 $s + t$ の形を含むとき, 状態遷移図を次のように更新する.

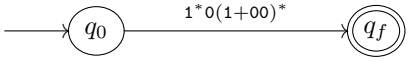


（手順 3） **$\varnothing$ と書かれた矢印を全て取り除いて** 得られる状態遷移図は, 正規表現 r が表す言語を認識する非決定性有限オートマトンである.

この様にして得られる ε-規則を許した非決定性有限オートマトンを $M[r]$ とおく.

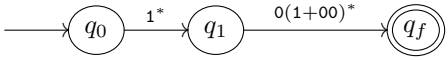
例 9-3　正規表現 $r = \mathtt{1}^*\mathtt{0}(\mathtt{1}+\mathtt{00})^*$ が表す言語を受理する ε 規則を許した非決定性有限オートマトン $M[r]$ を構成しよう.

(a) M_0 :



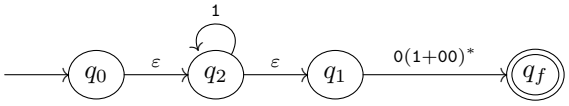
手順 1

(b) M_1 :



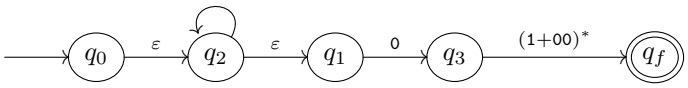
手順 2-2-2

(c) M_2 :



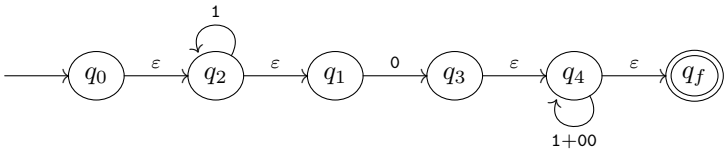
手順 2-2-1

(d) M_3 :



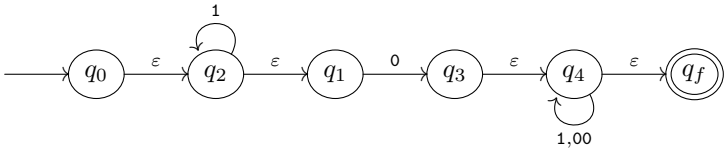
手順 2-2-2

(e) M_4 :



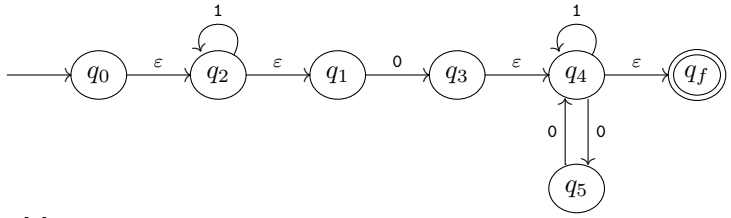
手順 2-2-1

(f) M_5 :



手順 2-2-3

(g) $M[r]$:



手順 2-2-2 （手順 3 は不要）

以上によって, $M[r]$ は,

$$K := \{q_0, q_1, q_2, q_3, q_4, q_5, q_f\}, \quad \Sigma := \{\mathtt{0}, \mathtt{1}\}, \quad F := \{q_f\}$$

であって, 開始状態が q_0, 遷移関数 $\delta : K \times (\Sigma \cup \{\varepsilon\}) \to 2^K$ としたときの 5 つ組 $(K, \Sigma, \delta, q_0, F)$ である.

δ	ε	0	1
$\to q_0$	$\{q_2\}$	$\varnothing$	$\varnothing$
q_1	$\varnothing$	$\{q_3\}$	$\varnothing$
q_2	$\{q_1\}$	$\varnothing$	$\{q_2\}$
q_3	$\{q_4\}$	$\varnothing$	$\varnothing$
q_4	$\{q_f\}$	$\{q_5\}$	$\{q_4\}$
q_5	$\varnothing$	$\{q_4\}$	$\varnothing$
$^*q_f$	$\varnothing$	$\varnothing$	$\varnothing$

小テスト 9-1　次の正規表現が表す言語に属しているものを全て選べ.

(1) $((0+1)(0+1)(0+1))^*$
(2) $(0+1)^*001(0+1)^*$
(3) $(0+\varepsilon)(1+\varepsilon)$
(4) $0(0+1)^*1+1(0+1)^*0$
(5) $\varnothing^*+(0+1)^*1$

小テスト 9-2　次の各文章で下線部の太字になっている箇所は誤りである. これを正しい文章に書き換えるとき, 太字部分の修正として最も適切なものを選べ

(1) ε-規則を許した非決定性有限オートマトン $M=(K,\Sigma,\delta,q_0,F)$ の遷移関数**$\delta : K \times (\Sigma \cap \{\varepsilon\}) \to K$** は, その入力として $K\times\Sigma$ の元だけではなく, $K\times\{\varepsilon\}$ の元も許している.
(2) r を任意の正規表現であるとすれば, $\varnothing$ と**連接演算**をとっても, それが表す言語を変えることはない.
(3) $r=(\mathtt{a}+\mathtt{b})^*(\mathtt{aa}+\mathtt{bbb})$ が表す言語を受理する ε-規則を許す非決定性有限オートマトン $M[r]$ には, **5 個の状態**が存在する.
(4) **言語 A または B が ε**であれば, その連接演算 $A\circ B$ は空集合になる.
(5) 2 つの正規表現 r と s について, これらの表す言語 $\mathcal{L}(r)$ と $\mathcal{L}(s)$ が等しいときに $r=s$ と定義すると, $0+10+111^*0=$ `0*1*0` である.

演習問題 9-1　$\Sigma=\{0,1\}$ 上の次の言語を表す正規表現を示せ.

(1) 奇数個の 1 を含む文字列全体
(2) 0 と 1 の個数が共に奇数であるような文字列全体

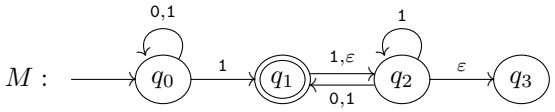
演習問題 9-2　$\Sigma=\{a,b\}$ 上の正規表現 $(\mathtt{a}+\mathtt{ab})^*\mathtt{a}^*$ と同じ言語を受理する ε-規則を許した非決定性有限オートマトン $M[r]$ を求めよ.

演習問題 9-3　2 つの正規表現 r と s について, これらの表す言語 $\mathcal{L}(r)$ と $\mathcal{L}(s)$ が等しいときに $r=s$ と定義する. 正規表現 r,s,t に対して, 次の等式を示せ.

(1) $r+s=s+r$　(2) $\varnothing+r=r$　(3) $\varnothing r=r\varnothing=\varnothing$
(4) $\varepsilon r=r\varepsilon=r$　(5) $r+r=r$　(6) $(r^*)^*=r^*$
(7) $(r+s)^*=(r^*s^*)^*=(r^*+s^*)^*$　(8) $(r+s)t=rt+st$　(9) $r(s+t)=rs+rt$

- **9-A ：ε-規則を許す非決定性有限オートマトン.**

ε-動作を許す非決定性有限オートマトンについて, もう少し詳しく述べる. ε-動作を許す非決定性有限オートマトン $M = (K, \Sigma, \delta, q_0, F)$ の各状態 $q \in K$ に対して, 状態 q と, q から ε と書かれた矢印のみを辿って到達する状態全体を **ε-閉包** と呼び, 記号で ε-CLOSE(q) とおく. 例えば



における各状態の ε-閉包は

$$\varepsilon\text{-CLOSE}(q_0) = \{q_0\}, \quad \varepsilon\text{-CLOSE}(q_1) = \{q_1, q_2, q_3\}, \quad \varepsilon\text{-CLOSE}(q_2) = \{q_2, q_3\}, \quad \varepsilon\text{-CLOSE}(q_3) = \{q_3\}$$

である. M の遷移関数 $\delta : K \times (\Sigma \cup \{\varepsilon\}) \to 2^K$ に対して, **拡張された遷移関数**

$$\widehat{\delta} : K \times \Sigma^* \longrightarrow 2^K$$

を次のように帰納的に定義する.

(i) $w = \varepsilon$ のとき, $\widehat{\delta}(q, \varepsilon) = \varepsilon\text{-CLOSE}(q)$.

(ii) $w = xa$ $(x \in \Sigma^*,\ a \in \Sigma)$ であるとする. このとき, $\widehat{\delta}(q, x) = \{r_1, r_2, \ldots, r_k\}$ であるとき,

$$\widehat{\delta}(q, w) = \varepsilon\text{-CLOSE}\left(\bigcup_{i=1}^{k} \delta(r_i, a)\right).$$

拡張された遷移関数 $\widehat{\delta}$ により, M が認識する言語は

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

と定義される.

ε-規則を許す非決定性有限オートマトンが認識する言語と ε-規則を許さない非決定性有限オートマトンが認識する言語には差がない. すなわち, 次の定理が示すことができる.

定理. 言語 L がある ε-規則を許す非決定性有限オートマトン M で認識されるための必要十分条件は, L がある ε-規則を許さない非決定性有限オートマトン M' で認識されることである.

証明. $M = (K, \Sigma, \delta, q_0, F)$ を ε-規則を許す非決定性有限オートマトンとすれば, M が認識する言語と同じ言語を受理する ε-規則を許さない非決定性有限オートマトン $N = (K, \Sigma, \delta', q_0, F')$ を以下のように構成する.

- $(q, a) \in K \times \Sigma$ に対して, $\delta'(q, a) := \widehat{\delta}(q, a)$ と定義する.
- $F' = \begin{cases} F & \text{if } \varepsilon\text{-CLOSE}(q_0) = \varnothing, \\ F \cup \{q_0\} & \text{if } \varepsilon\text{-CLOSE}(q_0) \neq \varnothing \end{cases}$

このとき, $\mathcal{T}(M) = \mathcal{T}(N)$ である. □