

無限語上のオートマトンで受理可能な 言語のクラスについて

山 崎 秀 記

1 前書き

コンピューター・システムのように休みなく動作し続けるシステムを考えると、そのようなシステムに対する入出力の列や、システムの動作の列を無限に続く記号列として捉えることができる。本論考では、無限の長さの記号列に対する基本的なシステム・モデルとして有限状態機械を考え、その能力の様々な特徴付けについて論じる。また、本論考では主要な結果の解説にとどめ、詳しい証明は割愛した。それらについては参考文献を参照してほしい。

無限長の記号列 (ω 語) に対する有限状態機械 (有限オートマトン) の理論は、Büchi によって、自然数上の単項 2 階述語論理 S1S の決定問題を解くために、始められた [1]。S1S の決定問題とは、与えられた論理式の真偽を判定する問題である。Büchi は、S1S の論理式 ψ を ω 語に対する条件を記述する式とみなし、 ψ を満たすような ω 語の集合を有限オートマトンで認識できること、逆に、有限オートマトンの動作が S1S の論理式で記述できることを示した。

これによって、S1S の論理式の真偽を判定する問題が、有限オートマトンが認識する ω 言語が空か否かという空集合判定問題に帰着され、後者の決定手続きを与えることにより、S1S の決定問題が決定可能であることが示される。

ところで、有限オートマトンで受理可能な ω 言語のクラスに付いては、S1S 論理式による特徴付けの他に、正則言語を用いて、

$$\bigcup_{i=1, n} K_i L_i^{\omega}$$

の形で表されることも知られている。本論考では、これらの特徴づけの能力の一部を制限することによって自然に導入される部分クラスの間に密接な対応があることを示す。

最近では、 ω 語の理論は計算機に関連する様々な分野で応用されている。例えば、プログラム言語の意味論、特に無限に続く計算の意味論が ω 語の枠組みの中で扱われる。また、平行的に動作するシステムの設計や解析と関連して、フェアな平行動作の記述にも ω 語の概念が使われている。

2 基本的定義

考えようとする記号列中に現れる記号を特定するために、例えば、システムへの入力を表す記号の集合や、出力を表す記号の集合、あるいはシステムの動作を表す記号の集合など、記号の集合を適当に定める。このように定められた記号の集合をアルファベットという。

アルファベット Σ に属す記号を並べたものを Σ 上の記号列という。記号列 α 中に現れる記号の総数を α の長さといい、 $|\alpha|$ で表す。例えば $\Sigma = \{a, b\}$ のとき、 $|aab| = 3$, $|babbaab| = 7$, $|bababa \dots| = \omega$ である。特に、長さ 0 の記号列を空列といい、 ε で表す。

Σ 上の記号列のうち、有限の長さの記号列を Σ 上の語、無限の長さの記号列を Σ 上の ω 語といい、 Σ 上の語全体の集合を Σ^* 、 ω 語全体の集合を Σ^ω と書く。さらに、 Σ^* の部分集合を Σ 上の言語、 Σ^ω の部分集合を ω 言語と呼ぶ。

記号列 α に対し、 α の n 番目の記号を $\alpha(n)$ と書く ($0 \leq n < |\alpha|$)。すなわち、

$$\alpha = \begin{cases} \alpha(0)\alpha(1)\dots\alpha(|\alpha|-1) & \alpha \in \Sigma^* \text{ のとき} \\ \alpha(0)\alpha(1)\dots & \alpha \in \Sigma^\omega \text{ のとき} \end{cases}$$

記号列 α, β にたいし、 α に続けて β をならべて得られる記号列を α と β の連接と言ひ、 $\alpha\beta$ と書く。正確には

$$\alpha\beta = \begin{cases} \alpha(0)\cdots\alpha(|\alpha|-1)\beta(0)\cdots\beta(1) & \alpha \in \Sigma^* \text{ のとき,} \\ \alpha & \alpha \in \Sigma^\omega \text{ のとき.} \end{cases}$$

任意の記号列 α は空列と接続しても変わらない。

$$\alpha\varepsilon = \varepsilon\alpha = \alpha.$$

また記号列 α を n 回接続してえられる記号列を α^n と書く。すなわち、

$$\alpha^0 = \varepsilon, \text{ かつ } \alpha^{n+1} = \alpha^n\alpha \quad (0 \leq n)$$

Σ 上の記号列の集合 $A, B \subseteq \Sigma^* \cup \Sigma^\omega$ に対し、 A と B の接続 AB を

$$AB = \{\alpha\beta \mid \alpha \in A, \beta \in B\}$$

と定義し、 A の n 乗 A^n を

$$A^0 = \{\varepsilon\}, \text{ かつ } A^{n+1} = A^n A \quad (0 \leq n)$$

と定義する。すなわち、 A^n は A に属する記号列を n 個接続した記号列の集合である。さらに

$$A^* = A^0 \cup A^1 \cup A^2 \cup \cdots$$

$$A^\omega = \{\alpha_0\alpha_1\cdots \mid \forall i \geq 0 \alpha_i \in A - \{\varepsilon\}\}$$

と定義すると、 A^* は A の記号列を有限個並べた記号列の全体、 A^ω は A の空列以外の記号列を無限個並べた ω 語の全体になる。

以後、表現を簡単にするために、 Σ 中の記号 a で、記号列 a だけからなる言語 $\{a\}$ を表すこととする。例えば、

$$(1) (ab)^* = \{\varepsilon, ab, abab, \cdots\}$$

は ab を有限回繰り返した語の集合、

$$(2) (a \cup ab)^* = \{\varepsilon, a, ab, aa, aab, aba, abab, \cdots\}$$

は空列かまたは a で始まり b が連続して現れないような語の集合、

$$(3) a^*b^\omega = \{bb\cdots, abb\cdots, aabb\cdots, aa\cdots abb\cdots, \cdots\}$$

は a を有限回繰り返した後 b が無限回続く ω 語の集合、

$$(4) (a^*b)^\omega = \{a^i b a^j b a^k b \cdots \mid i, j, k, \cdots \geq 0\}$$

は b が無限回現れる $\{a, b\}$ 上の ω 語の集合となる。

3 正則言語と ω 正則言語

第2節の最後の例題で与えたような、 Σ 上の記号と第2節で定義した接続と*演算、そして、集合の和を用いて表される言語を正則言語という。さらに正則言語をもとにして、集合の和、接続、 ω 演算を用いて表されるような ω 言語を ω 正則言語という。本節では、その正確な定義を与える。

Σ をアルファベットとする。 Σ 上の正則言語のクラスは次のように帰納的に定義される。

(1) 空集合 ϕ や、 Σ の部分集合は Σ 上の正則言語である。

(2) K, L が正則言語なら、 $K \cup L, KL, K^*$ は Σ 上の正則言語である。

さらに、 Σ 上の正則言語 $K_1, K_2, \dots, K_n, L_1, L_2, \dots, L_n$ を用いて次のように表される ω 言語を Σ 上の ω 正則言語という。

$$\bigcup_{i=1, n} K_i L_i^*$$

例えば、2節の最後にあげた例のうち、(1), (2) は正則言語、(3), (4) は ω 正則言語である。同様に例えば、 a と b のどちらか一方だけが無限回現れるような ω 語の集合も、 $\Sigma^* b^* \cup \Sigma^* a^*$ と書けるので、 ω 正則言語である。

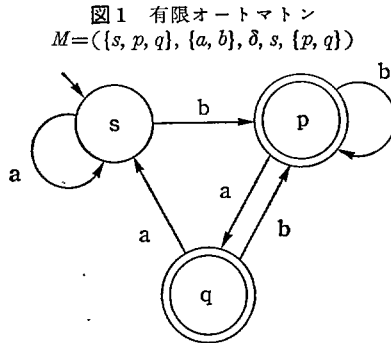
4 有限オートマトンと ω 正則言語

システムがとりうる状態の個数が有限であるような数学的モデルを有限オートマトンという。形式的には、アルファベット Σ 上の有限オートマトン M は次の5つのものから構成されるシステム $M = (Q, \Sigma, \delta, s, F)$ である。

- (1) 状態の有限集合 Q
- (2) アルファベット
- (3) 状態遷移関数 $\delta: Q \times \Sigma \rightarrow Q$
- (4) 初期状態 $s \in Q$
- (5) 受理状態の集合 $F \subseteq Q$

有限オートマトンは普通図1のように表示する。状態を丸で表し、状態遷移関数は Σ の元のラベル付きの矢印で表す。すなわち、 $\delta(p, a) = q$ のとき、状態

p から状態 q へラベル a をつけた矢印を引く. 初期状態は矢印で示し, 受理状態 (一つとは限らない) は 2 重丸で表す.



有限オートマトンはアルファベット Σ 上の記号列 α を入力として受け取り, 状態遷移関数によって定められる状態遷移の列 (Q 上の文字列) を引き起こす. それを $\text{run } M(\alpha)$ と書く. すなわち,

$$\text{run } M(\alpha)(0) = s,$$

$$\text{run } M(\alpha)(n+1) = \delta(\text{run } M(\alpha)(n), \alpha(n)) \quad (0 \leq n < |\alpha|)$$

例えば, 図1の有限オートマトンでは,

$$\alpha: \text{abbabaabbaba} \dots$$

$$\text{run } M: \text{ssppqqpsppqq} \dots$$

言い替えると, 有限オートマトンを表示した図の上で, 初期状態から出発して入力記号列 α にしたがって矢印を辿って行ったとき, 通過する状態の列が $\text{run } M(\alpha)$ である.

入力 α の長さが有限の場合には, 上で定めた $\text{run } M(\alpha)$ も Q 上の (有限長の) 語になり, α が ω 語ならば $\text{run } M(\alpha)$ も Q 上の ω 語になる.

入力が有限の長さである語の場合には, $\text{run } M(\alpha)$ に最後の状態が存在するので, それが指定された受理状態であるとき, 有限オートマトン M はその入力 α を受理するという. そして, M で受理される語の集合 $T(M)$ を M が受理する言語という. すなわち,

$$T(M) = \{w \in \Sigma^* \mid \text{run } M(w) \text{ の最後の状態が } F \text{ に属す}\}$$

例えば上図の有限オートマトンで受理される言語は $a^*b(a^*b)^*(\varepsilon \cup a)$ である。

有限オートマトンで受理される言語について次の定理が成り立つ [2, 7].

定理 1. Σ 上の言語 K が正則言語である必要十分条件は $K = T(M)$ となる有限オートマトン M が存在することである。

一方, 入力 α が ω 語の時は $\text{run } M(\alpha)$ も ω 語になるので, 語のときのように, 最後の状態というものがない。そこで, 次のような様々な受理条件が考えられている [2, 5, 6, 7].

(E) 受理状態が $\text{run } M(\alpha)$ に (少なくとも一度) 現れる。

(E') $\text{run } M(\alpha)$ に現れる状態はすべて受理状態である。

(L) 受理状態がすべて $\text{run } M(\alpha)$ に無限回現れる。

(L') ある受理状態が $\text{run } M(\alpha)$ に有限回しか現れない。

(I) ある受理状態が $\text{run } M(\alpha)$ に無限回現れる。

(I') $\text{run } M(\alpha)$ に無限回現れる状態はすべて受理状態である。

これらの受理条件のもとで有限オートマトン M で受理される ω 言語をそれぞれ, $E(M), E'(M), L(M), L'(M), I(M), I'(M)$ と書く。すなわち,

$$E(M) = \{\alpha \mid \text{run } M(\alpha) \text{ 中に } F \text{ の状態が現れる}\}$$

$$E'(M) = \{\alpha \mid \text{run } M(\alpha) \text{ 中に現れる状態はすべて } F \text{ に属す}\}$$

$$L(M) = \{\alpha \mid F \text{ の状態はすべて } \text{run } M(\alpha) \text{ 中に無限回現れる}\}$$

$$L'(M) = \{\alpha \mid \text{run } M(\alpha) \text{ 中に } F \text{ の状態が有限回現れる}\}$$

$$I(M) = \{\alpha \mid \text{run } M(\alpha) \text{ 中に } F \text{ の状態が無限回現れる}\}$$

$$I'(M) = \{\alpha \mid \text{run } M(\alpha) \text{ 中に無限回現れる状態はすべて } F \text{ に属す}\}$$

例えば図 1 の有限オートマトンでは

$$E(M) = a^*b\Sigma^*, \quad (b \text{ が一度は現れる } \omega \text{ 語の集合})$$

$E'(M) = \phi,$	(空集合)
$L(M) = (a^*bb^*a)^*$,	(a, b が無限回現れる ω 語の集合)
$L'(M) = \Sigma^*(a^* \cup b^*),$	(あるところから先は a だけかまたは b だけであるような ω 語の集合)
$I(M) = (a^*b)^*$,	(b が無限回現れる ω 語の集合)
$I'(M) = \Sigma^*(b \cup ab)^*$	(あるところからさきは b と ab だけをつなげてえられる ω 語の集合)

となる。

Σ 上の有限オートマトン M が存在して、 $E(M), E'(M), L(M), L'(M), I(M), I'(M)$ と表せる ω 言語のクラスをそれぞれ、 $\mathbf{E}, \mathbf{E}', \mathbf{L}, \mathbf{L}', \mathbf{I}, \mathbf{I}'$ と表す。実は、 ω 語上では有限オートマトンで受理できる ω 言語のクラスは、これらの受理条件によって異なることが知られている。

しかし、これらの受理条件では、有限オートマトンで ω 正則言語を受理できるだけの能力はない。有限オートマトンが任意の ω 正則言語を受理できる能力をもつためには、少し複雑な受理条件を与える必要がある。 \mathbf{F} を状態の集合をいくつか集めた集合とし、有限オートマトンの定義中の受理状態の集合 F の代わりに \mathbf{F} を指定したものを $M = (Q, \Sigma, \delta, s, \mathbf{F})$ を考え、受理条件を

(R) $\text{run } M(\alpha)$ 中に無限回現れる状態の集合が \mathbf{F} に属す。

とするのである。すなわち、

$R(M) = \{\alpha \mid \text{run } M(\alpha) \text{ 中に無限回現れる状態の集合が } \mathbf{F} \text{ の元}\}$

とする。 Σ 上の有限オートマトン M が存在して $R(M)$ と表せる ω 言語のクラスを \mathbf{R} と書く。このとき次の定理が成り立つ [2, 3, 4, 7]。

定理 2. Σ 上の ω 正則言語のクラスは \mathbf{R} と一致する。

受理条件を検討してみると ω 言語のクラス、 $\mathbf{E}, \mathbf{E}', \mathbf{L}, \mathbf{L}', \mathbf{I}, \mathbf{I}', \mathbf{R}$ の間には次のような関係が成り立つことがわかる [3, 6]。

定理 3.

- (1) $E' = \{A|\Sigma^\omega - A \in E\}$, $L' = \{A|\Sigma^\omega - A \in L\}$, $P' = \{A|\Sigma^\omega - A \in I\}$,
 (2) $E \subseteq L \subseteq I \subseteq R$, $E' \subseteq L' \subseteq P' \subseteq R$
 (3) $E \subseteq P'$, $E' \subseteq I$
 (4) $L \cap L' \subseteq E \cup E'$
 (5) (2), (3), (4) で示した以外の包含関係は成り立たない。

実際, $\Sigma = \{a, b\}$ とすると, 次のような例が知られている [6, 7].

$$a^\omega \in (L \cap E') - E$$

$$a^*b\Sigma^\omega \in (L' \cap E) - E'$$

$$b^\omega \cap b^*ab^\omega \in E' - L$$

$$ab^*a\Sigma^\omega \in E - L'$$

$$b^\omega \cup ab^*a\Sigma^\omega \in (L' \cap I) - E'$$

$$b^*ab^\omega \in (L \cap P') - E$$

$$\Sigma^*b^\omega \in L' - I$$

$$(b^*a)^\omega \in L - P'$$

$$ab^\omega \cup ba^*b\Sigma^\omega \in (I \cap P') - (L \cup L')$$

$$b\Sigma^*a^\omega \cup a\Sigma^*b^\omega \in P' - (I \cup L')$$

$$a(b^*a)^\omega \cup b(a^*b)^\omega \in I - (P' \cup L)$$

$$a\Sigma^*a^\omega \cup (ba^*)^\omega \in R - (I \cup P')$$

5 言語による ω 言語の表現

第3節では, ω 正則言語を次のように表される言語として定義した.

$$\bigcup_{i=1, n} K_i L_i^\omega$$

ただし, $K_1, K_2, \dots, K_n, L_1, L_2, \dots, L_n$ は Σ 上の正則言語である. また, クラス E に属す ω 言語に対しては, オートマトンが受理状態に達することが受理条件であるので, 次のような正則言語 K を用いて, $K\Sigma^\omega$ と書ける.

$$K = \{w \in \Sigma^* \mid \text{run } M(w) \text{ の最後の状態が受理状態で,}$$

途中には受理状態が現れない}

逆に、このような K を用いて $K\Sigma^{\circ}$ と表される ω 言語は E に属することも証明できる。本節では、有限オートマトンを用いて定義される様々なクラスが同様の特徴付けを持つことを示す。

記号列 α にたいし、その前半部分を α のプレフィックス (接頭語) という。すなわち、

$$\alpha(0)\alpha(1)\cdots\alpha(n) \quad (n < |\alpha|)$$

を α のプレフィックスという。特に、 α と異なる α のプレフィックスを真のプレフィックスという。 ($\alpha \in \Sigma^*$ の時、 α 自身も α のプレフィックスであるが、 $\alpha \in \Sigma^{\circ}$ のとき、 α 自身は α のプレフィックスではない。)

言語 K は、 K に属す語のプレフィックスが再び K に属することがないとき、プレフィックス・フリーであるという。プレフィックス・フリーで有限な Σ 上の言語のクラスを J 、プレフィックス・フリーな正則言語のクラスを P とする。

このとき、上でクラス E に属す ω 言語を表すのに用いた正則言語 K はプレフィックス・フリーな正則言語である。

有限オートマトンを用いて定義される Σ 上の ω 言語の様々なクラスに付いて、このような特徴付けが可能であり、次の定理が成り立つ [9].

定理 4.

- (1) $E \cap E' = \{J\Sigma^{\circ} \mid J \in J\}$
- (2) $E = \{P\Sigma^{\circ} \mid P \in P\}$
- (3) $L \cap E' = \{J_1 J_2^{\circ} \mid J_1, J_2 \in J\}$
- (4) $L \cap I' = \{PJ^{\circ} \mid P \in P, J \in J\}$
- (5) $L = \{P_1 P_2^{\circ} \mid P_1, P_2 \in P\}$
- (6) $I \cap I' = \{\cup_{i=1, n} P'_i (\Sigma^{\circ} - P_i^{\circ}) \mid P'_i, P_i \in P\}$
- (7) $I = \{\cup_{i=1, n} P'_i P_i^{\circ} \mid P'_i, P_i \in P\}$

また、定理 3 (1) に注意すれば、クラス E' , $L' \cap E$, $L' \cap I$, L' , I' もそれぞれ、定理 4 の (2), (3), (4), (5), (7) に属す ω 言語の補集合としての表現が可能で

ある。

6 ω 語上の位相

2つの ω 語 α, β の間の距離を $1/\min\{n+1|\alpha(n)\neq\beta(n)\}$ と定義すると Σ^ω に自然に位相が定義できる。この位相のもとでの、 Σ^ω 中の開集合は次のような ω 言語になる。[3, 6]

定理 5. (1) ω 言語 A が開集合である必要十分条件は $A=K\Sigma^\omega$ と書ける言語 $K(\subseteq\Sigma^\omega)$ が存在することである。

(2) ω 言語 A が開集合かつ閉集合である必要十分条件はある有限集合 F が存在して、 $A=F\Sigma^\omega$ と書けることである。

この位相によって Σ 上の ω 言語上にボレル階層が導入できる。すなわち、 ω 言語の中で上に定義した位相のもとで、開かつ閉な集合のクラス、開集合のクラス G 、閉集合のクラス F 、加算個の閉集合の和集合のクラス F_σ 、加算個の開集合の共通部分のクラス G_δ などが定義できる。なお、この上の階層、すなわち、加算個の F_σ 集合の共通部分のクラス $F_{\sigma\delta}$ や加算個の G_δ 集合の和集合のクラス $G_{\delta\sigma}$ は ω 正則言語のクラスを含んでしまうことが知られている。 ω 言語のクラス E, E', I, I' は位相的に次のように特徴付けられる [6]。

定理 6.

$$(1) \quad E \cap E' = F \cap G$$

$$(2) \quad E = R \cap G$$

$$(3) \quad E' = R \cap E$$

$$(4) \quad I = R \cap F_\sigma$$

$$(5) \quad I' = R \cap G_\delta$$

$$(6) \quad R \subseteq F_{\sigma\delta} \cap G_{\delta\sigma}$$

6 自然数上の単項2階述語論理 S1S

無限長の語 (ω 語) 上のオートマトンの研究は, Büchi により自然数上の単項2階述語論理 S1S の決定問題を解くために応用されたのが最初であった. S1S は自然数上を動く変数 x, y, \dots と ω 語上を動く変数 α, β, \dots を持ち, 原始論理式として, $t=u, t < u, \alpha(t) = a$ を持つ体系である. ここで, t, u は $x+n$ または n (n は自然数) という形の項である.

ここで, 自然数変数に対する限定記号 (\forall, \exists) と ω 語変数に対する限定記号 (\forall, \exists) を区別するために, 後者を太字で表すことにしよう. さて, 論理式中の自由変数を明確にするために, S1S の論理式 ψ が $x, y, \dots, \alpha, \beta, \dots$ を自由変数に持つとき, ψ を $\psi(x, y, \dots, \alpha, \beta, \dots)$ と書く. 論理式 $\psi(\alpha)$ に対し,

$$\{\alpha \mid \psi(\alpha) \text{ が成り立つ}\}$$

を ψ が定義する ω 言語という.

たとえば, $\Sigma = \{a, b\}$ のとき,

$$a^*b\Sigma^* \text{ は論理式 } \exists x \alpha(x) = b \text{ で定義され,}$$

$$(a^*bb^*a)^* \text{ は論理式 } \forall x \exists y \exists z (x \leq y \wedge x \leq z \wedge \alpha(y) = a \wedge \alpha(z) = b)$$

で定義される.

また, 有限オートマトン $M = (Q, \Sigma, \delta, s, F)$ を用いて $E'(M)$ と表せる ω 言語は

$$\exists \beta \forall x (\beta(0) = s \wedge [\delta(\beta(x), \alpha(x)) = \beta(x+1)] \wedge [\beta(x) \in F])$$

と書ける. $[]$ の中は必ずしも S1S の論理式の形をしていないが, S1S の限定記号を含まない論理式で書ける.

S1S の論理式で定義できる ω 言語のクラスについて次の定理が成り立つ [1].

定理 7. S1S で定義可能な ω 言語のクラスと ω 正則言語のクラスは一致する.

一般に, 任意の論理式は限定記号を前に出した形の論理式

$$Q_1\rho_1Q_2\rho_2\cdots Q_n\rho_n\psi$$

($\rho_1, \rho_2, \dots, \rho_n$ は自然数変数または ω 語変数, Q_1, Q_2, \dots, Q_n は対応する限定記号, ψ は限定記号を含まない論理式)

に変形できるが, そのような形の論理式を冠頭標準形の論理式という. SIS の論理式を冠頭標準形で表したとき, 前置部の形 (限定記号の列 $Q_1Q_2\cdots Q_n$) によって ω 正則言語のクラスを細かく分けることができる. すなわち, 限定記号の列 τ にたいし,

$$C(\tau) = \{A \mid A \text{ は冠頭標準形の前置部の限定記号の列が } \tau \text{ の形をした論理式で定義できる.}\}$$

例えば

$$C(\exists \forall) = \{A \mid A \text{ は } \exists \beta \forall x \psi(a, \beta, x) \text{ という冠頭標準形の論理式で定義される.}\}$$

したがって, 上に述べたことから, $E' \subseteq C(\exists \forall)$ がいえる. 実は, 逆の包含関係が成り立つことも, クラス E' の位相的特徴付けに注意するといえる.

このようにして, 論理式から導入された ω 正則言語のクラスが, 位相的に定義されたクラス, したがって有限オートマトンの受理条件によって導入されたクラスときれいに一致することが示せる [8]. ただし, \forall^+, \exists^+ などはいずれも限定記号 \forall, \exists が一個以上並んだ列, $\forall \forall^*, \exists \exists^*$ を表すものとする.

定理 8.

- (1) $E \cap E' = C(\tau)$ $\tau \in (\exists \cup \forall)^*$ のとき
- (2) $E = C(\tau)$ $\tau \in (\exists \cup \forall \cup \exists)^* \forall \exists^+ (\exists \cup \forall)^*$ のとき
- (3) $E' = C(\tau)$ $\tau \in (\exists \cup \forall \cup \forall)^* \exists \forall^+ (\exists \cup \forall)^*$ のとき
- (4) $I = (\tau)$ $\tau \in (\forall \cup \forall)^* \forall \forall^+ \exists^+ (\exists \cup \forall)^*$ のとき
- (5) $I' = C(\tau)$ $\tau \in (\exists \cup \exists)^* \exists \exists^+ \forall^+ (\exists \cup \forall)^*$ のとき
- (6) $R = C(\tau)$ $\tau \in (\exists \cup \forall \cup \exists \cup \forall)^* (\exists \forall^+ \exists \forall \cup \exists \forall^+ \forall \exists^+ \forall^+ \cup \exists \forall^+ \forall^+ \exists^+ \forall \forall \cup \exists \forall^+ \forall^+ \exists^+ \forall^+ \forall^+) (\exists \cup \forall)^*$ のとき

定理 8 より, τ が ω 語変数の限定記号を本質的に含むとき, クラス $C(\tau)$ は

E, E', I, P, R のどれかになることがわかる。 ω 語変数に対する限定記号を含まない1階の述語論理式で定義できるようなクラスに付いては、スター・フリーな ω 言語を用いた特徴付けが知られている [7]。

9 まとめ

本論考では ω 正則言語を定義づける3種類の方法を取り上げた。すなわち、

(1) 正則言語を用いて、

$$\bigcup_{i=1, n} K_i L_i^\omega$$

の形で表される ω 言語として、

(2) 有限オートマトンで受理される ω 言語として、

(3) 自然数上の単項2階述語論理式で定義される ω 言語として、

ω 正則言語は定義できる。

そして、位相的に定義される ω 正則言語の部分クラスおよび、上記の方法のそれぞれにおいて適当な制限を付けることによって自然に導入される ω 正則言語の部分クラスの間に、密接な関連があることを示した。

以下にそれを表の形でまとめてみよう。ただし、

(1) S1S 論理式の欄ではそのクラスを特徴付ける典型的な限定記号列のみ

オートマトン	位相	S1S	ω 正則表現
$E \cap E'$	$F \cap G$	ε	$J\Sigma^\omega$
$L \cap E'$			$J'J^\omega$
$L' \cap E$			$\overline{J'J^\omega}$
E	G	$\forall \exists$	$P\Sigma^\omega$
E'			$\overline{P\Sigma^\omega}$
$L \cap I'$	F	$\exists \forall$	PJ^ω
$L' \cap I$			$\overline{PJ^\omega}$
L			$P'P^\omega$
L'			$\overline{P'P^\omega}$
$I \cap I'$	$F_\sigma \cap G_\delta$		$\bigcup P'_i P_i^\omega$
I	F_σ	$\forall \exists \forall$	$\bigcup P'_i P_i^\omega$
I'	G_δ	$\forall \exists \exists$	$\bigcap \overline{P'_i P_i^\omega}$
R		$\forall \exists \forall, \forall \exists \forall$	$\bigcup K_i P_i^\omega$

を掲げ,

(2) ω 正則表現では, J は任意の有限プレフィックス・フリー言語を, P_i は任意のプレフィックス・フリー正則言語を, K_i は任意の正則言語を表し, また, $\Sigma^\omega - A$ を \bar{A} と表している.

10 参考文献

- [1] J. Buchi, On a decisions method in restricted second-order arithmetic, Internat. Congress on Logic, Methodology and Philosophy, Stanford, CA, 1960.
- [2] S. Eilenberg, Automata, Languages and Mchines, Vol. A (Academic Press, New York 1974).
- [3] L. Landweber, Decision problems for ω -automata, Math. Systems Theory 3 (1969) 376—384.
- [4] R. MacNaughton, Testing and generating infinite sequences by a finite automaton, Inform. and Control 9 (1966) 521—530.
- [5] T. Moriya and H. Yamasaki, Accepting conditions for ausomata on ω -words, Theoretical Computer Science 61 (1988) 137—147.
- [6] M. Takahashi and H. Yamasaki, A note on ω -rgeler languages, Theoretical Computer Science 23 (1983) 214—225.
- [7] 高橋正子, 言語と ω 言語, 数学 38 卷 4 号 (日本数学会, 1986) 302—316.
- [8] H. Yamasaki, M. Takahashi and K. Kobayashi, Characterization of ω -regular languages by monadic second-order formulas, Thoretical Computer Science 46 (1986) 91—99.
- [9] H. Yamasaki, Language-theoretical Comsentations of ω -languages, Theoretical repreputer Science 66 (1989) 247—254.

(一橋大学助教授)