数理解析研究所講究録 1375 巻 2004 年 221-228

SHシステムと拡張オートマトン
SH System and Extended Automaton

大阪府立大学理学系研究科 高石 理恵 * (Rie Takaishi)
大阪府立大学総合科学部 向内 康人 ** (Yasuhito Mukouchi)
大阪府立大学総合科学部 佐藤 優子 ** (Masako Sato)
*Graduate School of Sciences, Osaka Prefecture University
**College of Integrated Arts and Sciences, Osaka Prefecture University

要旨

本稿では, Mateescu et al. によって導入されたSHシステムとその言語について考察する. SHシステムは, Headによって導入されたDNA配列の組み換え, すなわちスプライシングモデルを簡素化したシステムである. SHシステムで生成される言語は, ある特別な性質を有する正規言語である. 本稿では, SHシステムと等価な言語生成能力を有するWord上のオートマトンを提示し, その等価性を示す. また, SHシステムで生成される言語の包含関係及び等価性の問題が, Word上のオートマトンやある有限集合間の問題に帰着することで解決されることを示す.

1 はじめに

昨今, ヒトゲノム計画等を通じて, 様々な生物種のDNA塩基配列解読が終わり, 焦点は解読の済んだ配列から構造や機能予測を行うための配列解析に移っている. その中でDNAからRNAに転写される際に行われる, 不要部分 (イントロン) の切り取りや配列の組み換え, すなわちスプライシングの部位 (site of splicing) を予測, 解析するために形式文法やマルコフモデル等の数理科学的手法が用いられている ([1], [3], [2], [4]).

DNA配列における組み換えモデルを導入したHead [1] では, スプライシングが起こる部位をスプライシングルールとして四つ組み $(u_1, u_2; u_3, u_4)$ で表現した. 与えられた配列はルールに従って組み換えられ, $x = x_1u_1u_2x_2$, $y = y_1u_3u_4y_2$ からは新しい配列 $v = x_1u_1u_4y_2$, $w = y_1u_3u_2x_2$ を得る.

SHシステム (Simple H System) は, 制限酵素によるDNA配列の切断から起こるスプライシング, 組み換えのモデルを簡易化したもので, Mateescu et al.[2] によって導入された.

本稿では, SHシステムで生成される言語とWord上のオートマトンで生成される言語の対応を考える. Word上のオートマトンは, 拡張されたオートマトンで, 通常の有限オートマトンが文字単位での状態遷移を考えるのに対し, Word上のオートマトンでは, 語 (文字列) を単位として状態遷移が定義される. 扱う言語の各語が有限種類の語の連接で構成されるような場合には, 通常の有限オートマトンで定義するよりもWord上のオートマトンで定義する方が自然であり, 種々の性質を解析する際にも有用であると考えられる.

まず, 与えられたSHシステムに対応するWord上のオートマトンの構成方法を示し, これらが生成する言語が一致することを示す. 一般に, 同じマーカーを用いる場合であっても, 同じ言語を定義するSHシステムは複数存在するが, これらを変換したWord上のオートマトンは同じものとなることが示され, 言語の等価性を考える上で有用である. また, 逆に, ある条件を満たすWord上のオートマトンが与えられた場合, それをもとに, 同じ言語を定義するSHシステムを構成できることも示される.

さらに, SHシステムが生成する言語については, 各マーカーを高々k個しか含まない語の集合S_kを用いて, 言語の有限性, 包含関係, 等価性などが特徴づけられることを示す.

2 準備

2.1 基本事項

この節では, 以降の議論に必要な言語に関する基本的な定義をまとめておく.

記号Nで非負整数全体の集合を表す. 記号Σでアルファベット (定数記号の有限集合) を表し, Σ^*でΣ上のすべての語の集合を表す. Σ^*の部分集合を, Σ上の言語という.

長さが0であるような語を空語 (または, 空列) と呼び, 記号εで表す. Σ^*から空語εを除いた語の集合をΣ^+で表す.

語$u, v \in \Sigma^*$の連接をuvまたは$u \cdot v$で表す. 言語$L_1, L_2 \subseteq \Sigma^*$に対して, 連接$L_1L_2$ (または, $L_1 \cdot L_2$) を$L_1L_2 = \{u_1u_2 \mid u_1 \in L_1,\ u_2 \in L_2\}$と定義する. また, 言語$L \subseteq \Sigma^*$に対して, $L^0 = \{\varepsilon\}$とし, 各$i \geq 0$に対して, $L^{i+1} = LL^i$とする. また, $L^* = \bigcup_{i \geq 0} L^i$, $L^+ = \bigcup_{i \geq 1} L^i$とする.

語$w \in \Sigma^*$に対して, wの接頭語, 部分語, 接尾語の集合$\mathrm{Pre}(w)$, $\mathrm{Sub}(w)$, $\mathrm{Suf}(w)$を

$$\begin{aligned}\mathrm{Pre}(w) &= \{v \in \Sigma^* \mid \exists u \in \Sigma^* \text{ s.t. } w = vu\},\\ \mathrm{Sub}(w) &= \{v \in \Sigma^* \mid \exists u, u' \in \Sigma^* \text{ s.t. } w = uvu'\},\\ \mathrm{Suf}(w) &= \{v \in \Sigma^* \mid \exists u \in \Sigma^* \text{ s.t. } uv = w\}\end{aligned}$$

と定める. 言語$L \subseteq \Sigma^*$に対して, $\mathrm{Pre}(L) = \bigcup_{w \in L} \mathrm{Pre}(w)$, $\mathrm{Sub}(L) = \bigcup_{w \in L} \mathrm{Sub}(w)$, $\mathrm{Suf}(L) = \bigcup_{w \in L} \mathrm{Suf}(w)$とおく. また, 語$w \in \Sigma^+$に対して, $\mathrm{tail}(w)$でwの最後の一文字を表す.

語$w \in \Sigma^*$に対して, $|w|$でwの長さを表し, 記号$a \in \Sigma$に対して, $|w|_a$でwに現れるaの数を表す. また, 記号の集合$S \subseteq \Sigma$と語$w \in \Sigma^*$に対して, $|w|_S = \sum_{a \in S} |w|_a$とおく.

集合Sに対して, Sの要素の数を$\sharp S$で表す. 集合S_1, S_2に対して, S_1とS_2の差$S_1 - S_2$を$S_1 - S_2 = \{x \mid x \in S_1,\ x \notin S_2\}$と定める.

2.2 SHシステム

定義 2.1 (Mateescu et al.[2]) **SHシステム** (Simple H System) とは, 3つ組$G = (\Sigma, M, A)$のこととする. ただし,

- Σ: アルファベット,
- M: マーカーの有限集合 ($M \subseteq \Sigma$),
- A: 公理 ($A \subseteq \Sigma^*$, 有限集合)

とする.

語 $x, y, z \in \Sigma^*$, $a \in M$ に対して, 関係 $(x, y) \vdash^a z$ を次のように定義する:

$$(x, y) \vdash^a z \Longleftrightarrow \exists x_1, x_2, y_1, y_2 \in \Sigma^* \text{ s.t. } x = x_1 a x_2,\ y = y_1 a y_2,\ z = x_1 a y_2.$$

言語 $L \subseteq \Sigma^*$ とマーカーの集合 M に対して, L からスプライシングで得られる言語 $\sigma_M^*(L)$ を次のように定義する:

$$\begin{aligned}
&\sigma_M(L) = L \cup \{z \in \Sigma^* \mid \exists x, y \in L, \exists a \in M \text{ s.t. } (x, y) \vdash^a z\},\\
&\sigma_M^0(L) = L,\\
&\sigma_M^{i+1}(L) = \sigma_M(\sigma_M^i(L)) \qquad (i \geq 0),\\
&\sigma_M^*(L) = \bigcup_{i \geq 0} \sigma_M^i(L).
\end{aligned}$$

SH システム $G = (\Sigma, M, A)$ が生成する言語 $L(G)$ を

$$L(G) = \sigma_M^*(A)$$

と定義する.

例 2.1 アルファベットを $\Sigma = \{a, b\}$, マーカーの集合を $M = \{a\}$, 公理を $A = \{ababba\}$ とする SH システム $G = (\Sigma, M, A)$ を考える.

このとき,

$$(ababba, ababba) \vdash^a aba, \qquad (ababba, ababba) \vdash^a abba$$

であるので, $aba, abba \in \sigma_M^*(A)$ である. また,

$$(aba, aba) \vdash^a ababa, \qquad (abba, abba) \vdash^a abbabba, \qquad (abba, aba) \vdash^a abbaba$$

であるので, $ababa, abbabba, abbaba \in \sigma_M^*(A)$ である. これらから,

$$L(G) = \sigma_M^*(A) = \{ab^{i_1}ab^{i_2}a \cdots ab^{i_k}a \mid k \geq 1,\ 1 \leq i_1, i_2, \cdots, i_k \leq 2\}$$

となることがわかる.

ここで, 公理を $A' = \{aba, abba\}$ とする SH システム $G' = (\Sigma, M, A')$ を考えても同じ言語を生成することがわかる.

この例からわかるように, 同じ言語を生成する SH システムは複数存在する. また, 次の例で示すように, 異なるマーカー集合を持つ 2 つの SH システムが同じ言語を生成する場合もあることに注意する.

例 2.2 $\Sigma = \{a, b, c\}$, $A = \{abacabaca\}$ とし, 2 つの SH システム $G = (\Sigma, \{b\}, A)$, $G' = (\Sigma, \{c\}, A)$ が生成する言語を考える.

このとき,

$$L(G) = \{a(baca)^i baca \mid i \geq 0\}$$

であり,

$$L(G') = \{aba(caba)^i ca \mid i \geq 0\}$$

であるが, 容易にわかるように, これらは同じ言語である.

2.3 Word 上のオートマトン

定義 2.2 **Word 上の非決定性有限オートマトン** (または, 単に **Word 上のオートマトン**) とは, 5 つ組 $Q=(K,W,\delta,q_0,F)$ のこととする. ただし,

- K : 有限状態集合,
- W : 語の集合 ($W\subseteq\Sigma^*$),
- δ : 状態遷移関数 ($K\times W\to 2^K$),
- q_0 : 初期状態,
- F : 最終 (受理) 状態 ($F\subseteq K$).

状態遷移関数 δ ををを次のように拡張する ($q\in K$, $w\in W^+$):

$$\begin{aligned}\delta(q,\varepsilon)&=\{q\},\\ \delta(q,w)&=\bigcup_{u\in \mathrm{Pre}(w)\cap W}\ \bigcup_{p\in\delta(q,u)}\delta(p,u\backslash w).\end{aligned}$$

ただし, $u\backslash w$ は, $w=uv$ となる v を表すものとする.

Word 上のオートマトン Q が受理する言語 $L(Q)$ を

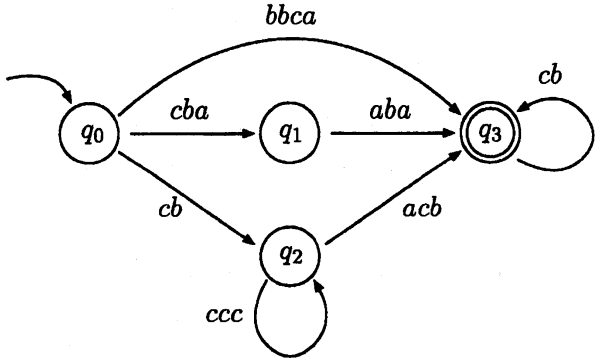
$$L(Q)=\{w\in\Sigma^*\mid \delta(q_0,w)\cap F\neq\phi\}$$

と定義する.

例 2.3 Word 上のオートマトン $Q=(K,W,\delta,q_0,F)$ を考える. ただし,

$$\begin{aligned}K&=\{q_0,q_1,q_2,q_3\},\\ W&=\{aba,acb,bbca,cb,cba,ccc\},\\ F&=\{q_3\}\end{aligned}$$

とし, 遷移関数 δ は, 次の図で示されるものとする.



このとき, $cbaaba$, $abcccacb$ は受理され, $bbcac$ は受理されない. 一般に,

$$L(Q)=\{bbca(cb)^j\mid j\geq 0\}\cup\{cbaaba(cb)^j\mid j\geq 0\}\cup\{cb(ccc)^iacb(cb)^j\mid i,j\geq 0\}$$

であることが示される.

3 SH システムとオートマトン

与えられた SH システム $G = (\Sigma, M, A)$ に対して, 次のように定義される Word 上のオートマトン $Q = (K, W, \delta, q_0, F)$ を対応させる:

- K : 状態集合

$$K = \{q_0\} \cup \{q_f\} \cup \{q_a \mid a \in M\}.$$

- $W \subseteq \Sigma^*$: 語の集合

$$W = W_{0f} \cup \bigcup_{a \in M} W_{0a} \cup \bigcup_{a,b \in M} W_{ab} \cup \bigcup_{b \in M} W_{bf}.$$

ただし, 各 $a, b \in M$ に対して,

$$\begin{aligned} W_{0f} &= \{w \mid w \in A \cap (\Sigma - M)^*\}, \\ W_{0a} &= \{wa \mid wa \in \mathrm{Pre}(A),\ w \in (\Sigma - M)^*\}, \\ W_{ab} &= \{wb \mid awb \in \mathrm{Sub}(A),\ w \in (\Sigma - M)^*\}, \\ W_{bf} &= \{w \mid bw \in \mathrm{Suf}(A),\ w \in (\Sigma - M)^*\} \end{aligned}$$

とする (記号 $0, f$ は M に属さない特別な記号とする).

- δ : 遷移関数 $(K \times W \to 2^K)$

各 $a, b \in M$ と $w \in W$ に対して,

$$\begin{aligned} q_f \in \delta(q_0, w) &\Longleftrightarrow w \in W_{0f}, \\ q_a \in \delta(q_0, w) &\Longleftrightarrow w \in W_{0a}, \\ q_b \in \delta(q_a, w) &\Longleftrightarrow w \in W_{ab}, \\ q_f \in \delta(q_b, w) &\Longleftrightarrow w \in W_{bf} \end{aligned}$$

とする.

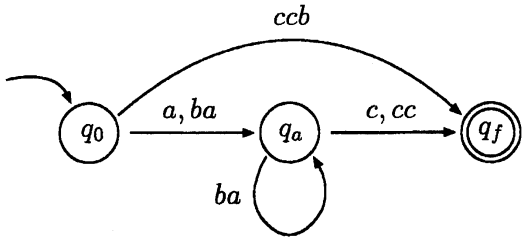
- q_0 : 初期状態

- $F = \{q_f\}$: 最終状態

例 3.1 SH システム $G = (\{a, b, c\}, \{a\}, \{abacc, babac, ccb\})$ に対応する Word 上のオートマトンは, $Q = (K, W, \delta, q_0, F)$ となる. ここで,

$$\begin{aligned} K &= \{q_0, q_a, q_f\}, \\ W &= W_{0f} \cup W_{0a} \cup W_{aa} \cup W_{af} = \{ccb\} \cup \{a, ba\} \cup \{ba\} \cup \{c, cc\} = \{ccb, a, ba, c, cc\}, \\ F &= \{q_f\} \end{aligned}$$

であり, 遷移関数 δ は, 次の図で示されるものである.



定理 3.1 G を SH システムとし, Q を G に対応する Word 上のオートマトンとする. このとき,

$$L(Q) = L(G)$$

が成立する.

一般に, 同じマーカー集合 M をもつ SH システムで同じ言語を生成するものは, 公理集合 A の与え方で複数存在するが, これらを Word 上のオートマトンに対応させたものは唯一であることがわかる.

定理 3.2 $G = (\Sigma, M, A)$, $G' = (\Sigma, M', A')$ を SH システムとし, Q, Q' を G, G' に対応する Word 上のオートマトンとする.

このとき, $L(G) = L(G')$ となるための必要十分条件は, $L(Q) = L(Q')$ である. 特に, $M = M'$ のときには, $Q = Q'$ となることが必要十分条件である.

この定理から, 与えられた 2 つの SH システムが同じ言語を生成するか否かが比較的簡単に判定できることがわかる.

定理 3.3 Word 上のオートマトン $Q = (K, W, \delta, q_0, F)$ が次の条件を満たすとする:

(1) $\forall q \in K, \forall w \in W, q_0 \notin \delta(q, w)$ (初期状態 q_0 への遷移を持たない),

(2) $\sharp F = 1$ (最終状態は 1 つである),

(3) $F = \{q_f\}$ とすると, $\forall w \in W, \delta(q_f, w) = \phi$ (最終状態からの遷移はない),

(4) $\forall q \in K, \forall w \in W, \sharp\delta(q, w) \leq 1$ (各状態 $q \in K$ から各語 $w \in W$ による遷移先は高々 1 つである),

(5) $M = \{\mathrm{tail}(w) \in \Sigma \mid w \in W,\ \exists q \in K - F \text{ s.t. } \delta(q, w) \neq \phi\}$ とすると, 語の集合 W と遷移関数 δ は, 次の条件を満たす:

(A) $W \subseteq (\Sigma - M)^* \cdot M \cup (\Sigma - M)^*$,

(B) $\forall w, w' \in (\Sigma - M)^* \cdot M, \forall q, q' \in K - F,$

$[\delta(q, w) \neq \phi,\ \delta(q', w') \neq \phi \Rightarrow (\mathrm{tail}(w) = \mathrm{tail}(w') \Leftrightarrow \delta(q, w) = \delta(q', w'))]$,

(C) $\forall w \in (\Sigma - M)^*, \forall q \in K - F,$

$[\delta(q, w) \neq \phi \Rightarrow \delta(q, w) \cap F \neq \phi]$.

ただし, Σ は $W \subseteq \Sigma^*$ を満たす最小のアルファベットとする.

このとき, $L(Q) = L(G)$ となるような SH システム G を構築できる.

定理 3.1で示した, SH システムに対応するオートマトンも上の定理の条件 (1) ～ (5) を満たすことに注意する.

4 SH システムで生成される言語間の包含関係, 等価性

この節では, SH システムで生成される言語間の包含関係, 等価性などの性質は, 次の有限集合で特徴づけできることを示す.

定義 4.1 SH システム $G = (\Sigma, M, A)$ に対して,

$$S_k(G) = \{w \in L(G) \mid \forall a \in M,\ |w|_a \leq k\} \qquad (k \geq 0)$$

とする.

次の命題から SH システムの公理は, 各マーカーを高々 2 個しか含まないと仮定してもよいことがわかる.

命題 4.1 $G = (\Sigma, M, A)$ を SH システムとする. このとき, SH システム $G' = (\Sigma, M, A')$ で, 次の条件を満たすものが存在する:

(1) $L(G) = L(G')$,

(2) $A' \subseteq S_2(G)$.

この定理の証明から, $L(G)$ が有限言語であるための必要十分条件は, $L(G) = S_1(G)$ となることもわかる.

SH システム $G = (\Sigma, M, A)$ に対して, 公理の集合 A は, $L(G)$ の有限部分集合であることから, 次の系が成立することになる.

系 4.2 $G = (\Sigma, M, A)$ を SH システムとする.

(1) $L(G)$ が空言語であるための必要十分条件は, A が空集合となることである.

(2) $L(G)$ が有限言語であるための必要十分条件は, $A \subseteq S_1(G)$ となる (すなわち, A の語は, 各マーカーを高々 1 回しか含まない) ことである.

上の (2) については, Mateescu et al.[2] で示されていることに注意する.

命題 4.3 $G = (\Sigma, M, A)$, $G' = (\Sigma, M, A')$ を同じマーカー集合 M を持つ 2 つの SH システムとする.

(1) $L(G) \subseteq L(G')$ となるための必要十分条件は, $S_2(G) \subseteq S_2(G')$ となることである.

(2) $L(G) = L(G')$ となるための必要十分条件は, $S_2(G) = S_2(G')$ となることである.

与えられた 2 つの SH システム G, G' から, 有限集合 $S_2(G), S_2(G')$ を生成することができ, これらを比較することでこれらの SH システムが生成する言語の包含関係, 等価性が判断できることになる.

さらに, マーカー集合 M と M' が異なる場合も含めた包含関係の必要十分条件を調べるため, いくつか補題を準備する.

語 $w \in \Sigma^*$ の部分語のうち, マーカー集合 M の記号を含まない部分語の集合 MSub を

$$\mathrm{MSub}(w, M) = \mathrm{Sub}(w) \cap (\Sigma - M)^*$$

とし, 言語 $L \subseteq \Sigma^*$ に対して, $\mathrm{MSub}(L, M) = \bigcup_{w \in L} \mathrm{MSub}(w, M)$ とする.

定理 4.4 $G = (\Sigma, M, A)$, $G' = (\Sigma, M', A')$ を SH システムとする. このとき, $L(G) \subseteq L(G')$ であるための必要十分条件は, 次の (1), (2) が成立することである:

(1) $\forall w \in S_3(G), \forall u \in \mathrm{MSub}(w, M'), \forall a \in M, |u|_a \leq 1$,

(2) $S_3(G) \subseteq L(G')$.

この定理から, G と G' のマーカー集合が異なる場合でも, $S_3(G)$ を生成して調べることで, $L(G) \subseteq L(G')$ となるかどうかが判定できることになる. (さらに, G と G' を入れ換えて適用すれば, $L(G) = L(G')$ となるかどうかが判定できる.)

5 おわりに

本稿では, SH システムがある条件をみたす Word 上のオートマトンと相互に変換可能であることを示した. また, 言語としての包含問題, 等価性についての条件を示した.

今後の課題として, 遺伝子データなどからそれらを生成する SH システムの効率的な推論問題などが考えられる.

参考文献

[1] T. Head, *Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviors*, Bulletin of Mathematical Biology 49 (1987) 737–759.

[2] A. Mateescu, Gh. Păun, G. Rozenberg, and A. Salomaa, *Simple splicing systems*, Discrete Applied Mathematics 84 (1998) 145–163.

[3] Gh. Păun and A. Salomaa, *DNA Computing based on the Splicing Operation*, Math. Japonica 43 (1996) 607–632.

[4] T. Yokomori, N. Ishida and S. Kobayashi, *Learning Local Languages and Its Application to Protein α-Chain Identification*, in Proceedings of the Twenty-Seventh Annual Hawaii International Conference on System Sciences (1994) 113–122.