2005－AL－103（3）
2005／11／11

1触手を持つコミュニケーティングPシステムについて

奥田剛, 井上克司, 井上敦之, 伊藤暁

山口大学工学部

あらまし　P システムは、生物学上の特徴をもつセルを用いた並列計算モデルであり, 本論文ではこのモデルを利用した 1 つの触手をもつ通信機能のある P システムを取り上げる. これを Communicating P System with One Tentacle (1CPST) と呼ぶ.

本研究では、次の結果を示す.

- 任意の有限オートマトン M が与えられたとき, M が受理する言語 L(M) を受理するような 1CPST を構成する方法.
- $T = \{0\}$ なる入力テープ集合は深さ 2 の 1CPST では受理できるが, 深さ 1 の 1CPST では受理できない.
- 任意の正則表現が与えられたとき, その正則表現を表す言語を直接受理するような深さ 2 の 1CPST を構成する方法.

On the Cmmunicating P System with One Tentacle

Gou OKUDA, Katsushi INOUE, Atsuyuki INOUE, Akira ITO

Faculty of Engineering, Yamaguchi University

Abstract　P system is a parallel computing model that uses the cell having the biolosical feature , and this paper investigates the P system that has the communicating function with one tentacle. It is called Communicating P System with One Tentacle (1CPST).

In this study, the following results are obtained.

- A method of constructing a 1CPST that accepts the same language as that of arbitrary finite automaton.
- It is impossible for 1CPST of depth 1 to accept the set $T = \{0\}$, which can be accepted by a 1CPST of depth 2.
- A method of constructing a 1CPST's of depth 2 that directly accepts the languages expressed by arbitrary regular expressions.

1 まえがき

P システムは, 生物学上の特徴をもつセルを用いた並列計算モデルとして G.Paun によって提唱された [1]. このモデルは一番外側の skin membrane の中に階層性のある薄膜により構成されている. P システムの精密な定義は文献 [2] に記述されている. P システムの様々なモデルは幅広く研究されてきている. 例えば, 文献 [3] では P システムや様々な計算モデルのシステムに焦点を当てて, 計算の複雑さについて言及している. 現在までに様々な１次元言語上の P システムに関する計算モデルの研究がされ, その性能はチューリングマシンと同等であることが示されている. 文献 [4] では, １つのヘッド (触手) をもつ通信機能のある P システムや P システムに制限を加えたシステムが提唱されている (触手をもつシステムとしては, 文献 [5] で初めて言及されている). 前者は Communicating P System with One Tentacle (1CPST) と呼ばれるモデルであり, 後者は Restricted model of Communicating P System (RCPS) と呼ばれるモデルである. CPST では, ある言語 L がマルチヘッド有限オートマトンで受理される場合, その言語 L で受理されるような CPST が存在すること, 正の整数 r に対して, ある言語 L が $r+1$ 個の触手をもつ CPST で受理されるが,r 個の触手をもつ CPST では受理することが出来ないことなどが知られている [4]. 1CPST は言語受理能力において有限オートマトンと同等であることが示されている. 本論文では, 1CPST に関して, 次の 2 点について考察する.

1. 任意の有限オートマトンを模倣するのに必要な 1CPST の薄膜構造の深さはどの程度であるか.

2. 任意の正則表現が与えられたとき, その正則表現が表す言語を受理する 1CPST をどのように構成するか.

2 諸定義と諸記法

2.1 決定性（非決定性）有限オートマトン

決定性有限オートマトン（deterministic finite automaton）は, 次に示すような２つの有限集合 Q, Σ とこれらの関係を定める関数 δ, 特定の状態 q_0, および特定の状態集合 F を指定することにより定まる５項組（5-tuple）のシステム $M = (Q, \Sigma, \delta, q_0, F)$ として定義される.

1. 状態の有限集合 Q,

2. 入力記号の有限集合 Σ,

3. 状態推移関数 δ: 現在の状態 $q \in Q$ とそれへの入力 $a \in \Sigma$ の全ての組み合わせに対して, 次の時点にとるべき状態 $q' \subseteq Q$ を $\delta(q, a) = (q', d)$ により一意的に定める規則,

4. $q_0 \in Q$:初期状態,

5. $F \subseteq Q$:受理状態の集合.

また, 非決定性有限オートマトン（nondeterministic finite automaton）は上記の５項組のシステム M において状態遷移関数 δ を以下のように変更することで定義される.

3. 状態推移関数 δ: 現在の状態 $q \in Q$ とそれへの入力 $a \in \Sigma$ の全ての組み合わせに対して, 次の時点にとるべき状態の集合 $Q_i \subseteq Q$ を $\delta(q, a) = (Q_i, d)$ により一意的に定める規則,

ここに, $\delta(q, a) = (q', d)$ は, M の制御状態が q で入力ヘッドの読む記号が a のときは, M は１ステップの動作で, 制御部の状態を q' に変え, 入力ヘッドを d の方向（$d = -1$ のときは左方向, $d = +1$ のときは右方向, $d = 0$ のときは静止）に動かすことができることを意味する.

決定性有限オートマトンを DFA, 非決定性有限オートマトンを NFA と記す.

2.2 決定性（非決定性）有限オートマトンによる受理

x を決定性オートマトン $M = (Q, \Sigma, \delta, q_0, F)$ への入力語とする. M が初期状態 q_0 から出発し, x を読んでいくとき, 制御部の状態が F に到達したならば, x は M で受理されるという.

また, 非決定性オートマトン $M = (Q, \Sigma, \delta, q_0, F)$ への入力語とする. M が初期状態 q_0 から出発し, x を読んでいくとき, いつか制御部の状態が F に含まれるような１つの動作系列が存在するならば, x は M で受理されるという. M で受理されるすべての入力語の集合を $L(M)$ と記す.

2.3　P System

P system G はツリー構造のような薄膜から成っている．そして, 薄膜によって囲まれた空間を region と呼ぶ．１つの region は各々の薄膜に関係づけられる．一番外側の薄膜は skin membrane, 中に別の薄膜を含んでいない薄膜は elementary membrane と呼ばれ, skin membrane の外側の領域は environment と呼ばれる．従って, environment も含めて, n 個の薄膜をもつシステムは $n+1$ 個の region をもつ．各薄膜にはルールが設定される (無い場合もある)．この進化ルールは以下のような型で表現される．V とはシステムで使われる全てのオブジェクトの集合である．

1. $a \to a_t$,
2. $ab \to a_{t1}b_{t2}$,
3. $ab \to a_{t1}b_{t2}c_{come}$,

   $a, b, c \in V, \quad t, t1, t2 \in (\{here, out\} \cup \{in_j \mid 1 \le j \le n\}), 1 \le i \le n.$

オブジェクトの下に書いてある out（in_j）は, そのオブジェクトは現在含まれてる薄膜の外に直ちに移動すること（オブジェクトに隣接している j とラベル付けされた薄膜の中に直ちに移動すること）を意味する．オブジェクトの下に書いてある $here$ は, オブジェクトが移り変わり後も同じ薄膜内に残ることを意味する．ここで, (3) の進化ルールの型は, skin membrane によって囲まれた region でのみ適用可能である．(3) の進化ルールの型が適用された時, environment からオブジェクト c が skin membrane に移動し, この region の要素となる．各ステップで進化ルールは maximally parallel manner（極大並列的）に適用される．maximally parallel manner とは適用可能なルールをまとめて実行する方法である．environment には豊富なオブジェクトが存在する．最初に environment のオブジェクトは利用できないが, 計算途中で environment に取り入られるかもしれない．このシステムはオブジェクトを全ての薄膜に分配した状態を初期状態として固定して動き出し, 適用できる進化ルールがなくなった時に停止する．図１に P system の例を示す．ここで, skin membrane は 1 の membrane のことであり, elementary membrane は 2,3,5,7,8,9 の membrane のことである．この図の場合, 9 個の membrane をもっているので, 10 個の region をもつ．

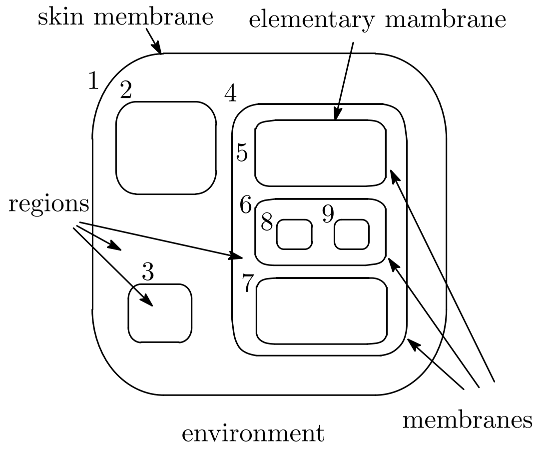

図 1: P system の例

2.4　Communicating P System with One Tentacle（1CPST）

ここで P system に１つの触手の付いたシンプルなモデルを考える．これを 1CPST と呼ぶ．1CPST G は Σ^* から成る終端記号の付いた２方向入力テープを読み込むものとする．Σ は V の部分集合で構成されており, 入力は決して変わらない．G は１つの触手 t をもつ．その触手は初期状態で左終端記号を指している．触手は左右に移動することによりテープを読み込む．そして, 次のように仮定する．薄膜は Σ の記号のオブジェクトは持ってはいけない．つまり, 使用できるオブジェクトは $V - \Sigma - \{\not{c}, \$\}$ である（¢ は左境界記号, $ は右境界記号).
1CPST は以下のような構造で定義される．

$$G = (V, \Sigma, \mu, w_1, \cdots, w_n, R_1, \cdots, R_n),$$

1. V はアルファベット．（この要素をオブジェクトと呼ぶ.）
2. $\Sigma \subseteq V$ は入力アルファベット．
3. μ は薄膜を $1, 2, \ldots, n$ とラベル付けされた, 薄膜構造を記号で表したもの．ここで, n を G の degree と呼ぶ．
4. $w_i (1 \le i \le n)$ は初期状態のオブジェクトの配置を記述している．
5. $R_i (1 \le i \le n)$ は各薄膜の進化ルールであり, 以下の型に限られる．

   (1) $a \to a_t$,

   (2) $ab \to a_{t1}b_{t2}$,

(3) $ab \to a_{t1}b_{t2}\sigma_{come,d}$.

$a, b \in V - \Sigma - \{¢, \$\}, \sigma \in (\Sigma \cup \{¢, \$\})^*$
$t, t1, t2 \in \{here, out\} \cup \{in_j \mid 1 \leq j \leq n\}$,
$d = \{+1, -1\}$

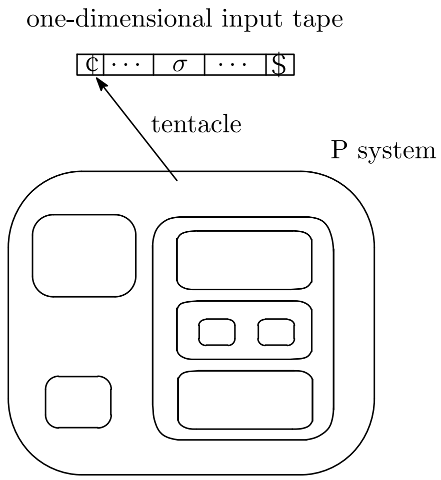

図 2: 1CPST の例

(3) の進化ルールの型は, skin membrane によって囲まれた region でのみ適用可能である．(3) の進化ルールの型が適用された時, environment からオブジェクト σ が skin membrane に移動し, この region の要素となる．そして, d の値 (+1, −1) により触手は入力テープ上を右, 左に動かす．σ というオブジェクトは skin membrane の region に残される．そして, 他の region に移動することはない．もし, 触手の指している入力テープの記号が σ ではない場合, 進化ルールは適用できない．触手に適用できる進化ルールが 2 つ以上ある時には非決定的に選択される．G が停止した時, 入力テープは受理されるとする．ここで, G で受理されるすべての言語（入力語の集合）を $L(G)$ と記す．全ての入力語に対して G の計算が 1 つしか存在しない場合, G を決定性 1CPST と呼ぶ．一般的に G は非決定性である．ここで r を正数とする．薄膜構造を表現する μ の深さが r の場合, G を深さ r の 1CPST と呼ぶ（μ が skin membrane だけの場合, 深さ 1 と定義する）．図 2 に 1CPST の例を示す．図 1 の P system の例に tentacle と one-dimensional input tape を付け足したものである．

決定性1CPST（非決定性1CPST）を1DCPST（1NCPST）と記す．そして, 深さ r の決定性 1CPST（非決定性 1CPST）を 1DCPST(r)（1NCPST(r)）と記す．1DCPST（1NCPST, 1DCPST(r), 1NCPST(r)）で受理される全ての言語の集合を£[1DCPST]（£[1NCPST], £[1DCPST(r)], £[1NCPST(r)]）と記す．例えば, £[1DCPST] とは 1DCPST G が受理する言語族を意味する．

3　任意の有限オートマトンを模倣するのに必要な1CPSTの薄膜構造の深さ

本章では任意の有限オートマトンを模倣するのに必要な 1CPST の薄膜構造の深さについて考察する．まず初めに, 1DCPST (1NCPST) と DFA (NFA) の受理能力の関係を示す．

補題 3.1　各 $r \geq 1$, $X \in \{\mathrm{D}, \mathrm{N}\}$ に対して,

$$£[1X\mathrm{CPST}(r)] \subseteq £[X\mathrm{FA}].$$

証明　$1X$CPST(r) のオブジェクトおよびルールの数は有限なので, 有限状態機械 XFA で模倣できることは明らかである．□

補題 3.2　各 $X \in \{\mathrm{D}, \mathrm{N}\}$ に対して,

$$£[X\mathrm{FA}] \subseteq £[1X\mathrm{CPST}(2)].$$

証明　任意の有限オートマトン M を以下のように定義する．

$$M = (Q, \Sigma, \delta, q_0, F),$$

ここに, $Q = \{q_0, q_1, q_2, \cdots, q_n\}$ とする ($n \geq 1$)．このとき, 1CPST(2) G は以下のように定義する．

$$G = (V, \Sigma, \mu, w_{skin}, w_0, w_1, \cdots, w_n, w_{loop}, R_{skin}, R_0, R_1, \cdots, R_n, R_{loop}),$$

1. $V = Q \cup \Sigma \cup \{c\}$,
2. $\mu = [_{skin}[_0]_0\ [_1]_1\ \cdots\ [_n]_n\ [_{loop}]_{loop}]_{skin}$,
3. $w_{skin} = \{c, q_0\}, w_0 = \lambda, w_1 = \{q_1\}, \cdots,$ $w_n = \{q_n\}, w_{loop} = \lambda$,
4. 各 $q_i, q_j \in Q, i \in \{0, \ldots, n\}, j \in \{0, \ldots, n\}, \sigma \in \Sigma \cup \{¢, \$\}, d \in \{+1, 0, -1\}$ に対して,

(1) $\delta(q_i, \sigma) \ni (p_j, d) \iff cq_i \to c_{in,j}q_{i,in,j}\sigma_{come,d} \in R_{skin}$,

(2) $\delta(q_i, \sigma) = \phi \iff$
$$\begin{cases} \text{もし } q_i \notin F \text{ ならば,} \\ cq_i \to c_{in,loop} q_{i,in,i} \sigma_{come,0} \in R_{skin}, \\ \text{もし } q_i \in F \text{ ならば,} \\ cq_i \to c_{out} q_{i,out} \sigma_{come,0} \in R_{skin}, \end{cases}$$

(3) $R_0 = \{cq_0 \to c_{out} q_{0,out}\}, \ldots, R_n = \{cq_n \to c_{out} q_{n,out}\}$,

(4) $R_{loop} = \{c \to c_{here}\}$.

$1X$CPST(2) G の動作を説明する.

- $1X$CPST(2) G の初期状態について：
初期状態 q_0 を skin membrane に配置する. また適当な記号（ここでは c ）も配置する. そして, q_1 を membrane 1, ..., q_n を membrane n に配置する. これを 1XCPST(2) G の初期状態とする.

- $1X$CPST(2) G の進化ルールについて：
(1) の定義によって, c は次の状態 q_j に対応するオブジェクトのある membrane j に入り, 現在の状態 q_i は membrane i に格納される. そして, (3) の定義により, 次の状態に移行したときにその状態が skin membrane に出される. (2) の定義によって, 次の状態がない（入力テープの最後を読んだ場合の）ときの, M の受理, 不受理状態を示している.
  - M が受理状態の場合, $1X$CPST(2) G は停止状態になる必要があるので, 各進化ルールに共通している c のオブジェクトを skin membrane の外に出して $1X$CPST(2) G は停止する.
  - M が不受理状態の場合, $1X$CPST(2) G はループ状態になる必要があるので, そのためにループさせる為の membrane (membrane loop と呼ぶ) を作り, そこに c を入れる. (4) の定義により, ループを実現している.

以上より, XFA と同等の受理能力をもつ $1X$CPST(2) を構成できるのは明らかである. □

補題 3.1 と補題 3.2 より次の定理を得る.

定理 3.1 各 $r \geq 2$, $X \in \{\mathrm{D}, \mathrm{N}\}$ に対して,

$$\mathcal{L}[1X\mathrm{CPST}(r)] = \mathcal{L}[1X\mathrm{CPST}(2)] = \mathcal{L}[X\mathrm{FA}].$$

また, 定理 3.1 と $\mathcal{L}[\mathrm{DFA}] = \mathcal{L}[\mathrm{NFA}]$ より次の系を得る.

系 3.1 各 $r \geq 2$ に対して,

$$\mathcal{L}[1\mathrm{DCPST}(r)] = \mathcal{L}[1\mathrm{NCPST}(r)].$$

次に $r = 1$ の場合を調べる.

定理 3.2 $\mathcal{L}[\mathrm{DFA}] - \mathcal{L}[1\mathrm{DCPST}(1)] \neq \emptyset$.

証明 $T = \{0\}$ なる入力テープを考える. T を受理するDFA を設計できるのは明らかである. 1DCPST(1) G が T を受理できないことは, 背理法によって証明できる. 詳細については省略する. □
定理 3.1 と定理 3.2 より次の系を得る.

系 3.2 $\mathcal{L}[1\mathrm{DCPST}(1)] \not\subseteq \mathcal{L}[1\mathrm{DCPST}(2)]$.

4 正則表現が表す言語を受理する 1CPST の構成法

本章では任意の正則表現が与えられたとき, その正則表現が表す言語を受理する 1CPST の構成法について考察する. ここでは, 1CPST を次のように定義する.
1CPST は触手を常に右だけに移動させるような 1 方向 1CPST を考える. つまり, 進化ルールの触手の動きを示す d は常に +1 である. また, 係って入力テープとしては左終端記号 ¢, 右終端記号 \$ を除外したテープ[1] を考える. 終端記号のない 1 方向 1CPST G は, 入力記号列の 1 番左側のオブジェクトに触手がある状態から動作を開始する. そして, 1 方向 1CPST G は, 入力記号列 w 上での計算で触手が w からはみ出して停止したとき, w は受理されたものとする. また空記号列 ϵ が受理されるというのは, 触手を動かさないで有限ステップ後に停止することと定義する.

正則表現は以下の規則よって定義される.

1. ϕ を, 空集合を表す正則表現とする.

2. ϵ を, $\{\epsilon\}$ を表す正則表現とする.

3. a を, $\{a\}$ $(a \in \Sigma)$ を表す正則表現とする.

[1] 構成の都合上境界記号をつけていない. 境界記号があると複雑な正則表現に対して, 以下の規則を繰り返し適用させる度に境界記号が付属する為に都合が悪いためである.

4. E_1, E_2 を言語 L_1, L_2 を表す正則表現とし, E_1+E_2(和), E_1E_2(連接), $E_1{}^*$(スター閉包) を言語 $L_1 \cup L_2$, L_1L_2, $L_1{}^*$ を表す正則表現とする.

定理 4.1 正則表現 E から直接（有限オートマトンを介さないで）, $L(E)$ を受理する境界記号のない 1 方向 1NCPST(2) を構成することができる.

証明 正規言語の各演算 E に対して, 1 方向 1NCPST(2) G を構成する. ここで, G は停止させるための membrane (accepting membrane と呼ぶ) を持つ. またここで使われる c というオブジェクトは $cursor$ を意味する特別なオブジェクトである.

基底段階：

Case B-1　$E = \phi$ のとき：境界記号のない 1 方向 1NCPST(2) G を次のように定義する.

$G = (\{c\} \cup V, \Sigma, \mu, w_0, w_{init}, w_{ac}, R_0, R_{init}, R_{ac})$, ここで,

1. $V = \{q_0, q_f\} \cup \Sigma$
2. $\mu = [_0[_{init}]_{init}[_{ac}]_{ac}]_0$
   $[_{init}]_{init}$: initial membrane (初期薄膜)
   $[_{ac}]_{ac}$: accepting membrane (受理薄膜)
3. $w_0 = \epsilon, w_{init} = cq_0, w_{ac} = q_f$
4. (1) $R_0 = \{cq_0 \to c_{here}q_{0,here}\}$
   (2) $R_{init} = \{cq_0 \to c_{out}q_{0,out}\}$
   (3) $R_{ac} = \phi$.

これにより, どんな入力記号列に対しても G は決して停止することがない. つまり, $L(G) = \phi$.

Case B-2　$E = \epsilon$ のとき：境界記号のない 1 方向 1NCPST(2) G を次のように定義する.

$G = (\{c\} \cup V, \Sigma, \mu, w_0, w_{init}, w_{ac}, R_0, R_{init}, R_{ac})$, ここで,

1. $V = \{q_0, q_f\} \cup \Sigma$
2. $\mu = [_0[_{init}]_{init}[_{ac}]_{ac}]_0$
   $[_{init}]_{init}$: initial membrane (初期薄膜)
   $[_{ac}]_{ac}$: accepting membrane (受理薄膜)
3. $w_0 = \epsilon, w_{init} = cq_0, w_{ac} = q_f$
4. (1) $R_0 = \{cq_0 \to c_{in_{ac}}q_{0,in_{init}}\}$
   (2) $R_{init} = \{cq_0 \to c_{out}q_{0,out}\}$
   (3) $R_{ac} = \phi$.

初めに, G は 4.(2) の進化ルールを適用することにより, initial membrane $[_{init}]_{init}$ からオブジェクト cq_0 を skin membrane $[_0]_0$ に追い出す. そして, 4.(1) の進化ルールを適用することにより, skin membrane $[_0]_0$ から $cursor$ オブジェクト c を accepting membrane $[_{ac}]_{ac}$ の中に追い出す. こうすることにより G は触手を右に動くことなく停止する. つまり, $L(G) = \{\epsilon\}$.

Case B-3　$E = a$ $(a \in \Sigma)$ のとき：境界記号のない 1 方向 1NCPST(2) G を次のように定義する.

$G = (\{c\} \cup V, \Sigma, \mu, w_0, w_{init}, w_{ac}, R_0, R_{init}, R_{ac})$, ここで,

1. $V = \{q_0, q_f\} \cup \Sigma$
2. $\mu = [_0[_{init}]_{init}[_{ac}]_{ac}]_0$
   $[_{init}]_{init}$: initial membrane (初期薄膜)
   $[_{ac}]_{ac}$: accepting membrane (受理薄膜)
3. $w_0 = \epsilon, w_{init} = cq_0, w_{ac} = q_f$
4. (1) $R_0 = \{cq_0 \to c_{in_{ac}}q_{0,in_{init}}a_{come,1}\}$
   (2) $R_{init} = \{cq_0 \to c_{out}q_{0,out}\}$
   (3) $R_{ac} = \phi$.

初めに, G は 4.(2) の進化ルールを適用することにより, initial membrane $[_{init}]_{init}$ からオブジェクト cq_0 を skin membrane $[_0]_0$ に追い出す. そして, 4.(1) の進化ルールを適用することにより, skin membrane $[_0]_0$ から $cursor$ オブジェクト c を accepting membrane $[_{ac}]_{ac}$ の中に追い出し, a を触手から skin membrane $[_0]_0$ に取り入れられ, 触手を右に 1 つ分動かす. こうして G は停止する. つまり, $L(G) = \{a\}$.

帰納段階：

Case I-1　$E = E_1 + E_2$ のとき：境界記号のない 1 方向 1NCPST(2) G_1, G_2 を次のように定義する.

$G_1 = (\{c\} \cup V_1, \Sigma_1, \mu_1, w_{01}, w_{init1}, w_{11}, \ldots, w_{1k_1}, w_{ac1}, R_{01}, R_{init1}, R_{11}, \ldots, R_{1k_1}, R_{ac1})$

$G_2 = (\{c\} \cup V_2, \Sigma_2, \mu_2, w_{02}, w_{init2}, w_{21}, \ldots, w_{2k_2}, w_{ac2}, R_{02}, R_{init2}, R_{21}, \ldots, R_{2k_2}, R_{ac2})$

$L(G_1) = L(E_1), L(G_2) = L(E_2)$ である. ここに,

1. $k_1, k_2 \geq 0$
2. $\mu_1 = [_{01}[_{init1}]_{init1}[_{11}]_{11} \ldots [_{1k_1}]_{1k_1}[_{ac1}]_{ac1}]_{01}$
   $[_{init1}]_{init1}$: initial membrane (初期薄膜)
   $[_{ac1}]_{ac1}$: accepting membrane (受理薄膜)

3. $\mu_2 = [_{02}[_{init2}]_{init2}[_{21}]_{21} \cdots [_{2k_2}]_{2k_2}[_{ac2}]_{ac2}]_{02}$
   $[_{init2}]_{init2}$: initial membrane (初期薄膜)
   $[_{ac2}]_{ac2}$: accepting membrane (受理薄膜)

4. − $w_{01} = \epsilon, w_{init1} = cq_{01}, w_l = q_l$ $(l \in \{11, \ldots, 1k_1\}), w_{ac1} = q_{f1}$,
   − $R_{init1} = \{cq_{01} \to c_{out}\, q_{01\,out}\}, R_{ac1} = \phi$,
   − $w_{02} = \epsilon, w_{init2} = cq_{02}, w_l = q_l$ $(l \in \{21, \ldots, 2k_2\}), w_{ac2} = q_{f2}$,
   − $R_{init2} = \{cq_{02} \to c_{out}\, q_{02\,out}\}, R_{ac2} = \phi$.

ここで, 1 方向 1NCPST(2) G_1, G_2 の V_1, V_2 をばらし, membrane の配置をばらし, それらの呼び名を以下のように変更する. q_0, q_f は新しいオブジェクトである.

$$G = (\{c\} \cup \{q_0, q_f\} \cup V_1 \cup V_2, \Sigma_1 \cup \Sigma_2, \mu, w'_0, w'_{init}, w'_{init1}, w'_{11}, \ldots, w'_{1k_1}, w'_{ac1}, w'_{init2}, w'_{21}, \ldots, w'_{2k_2}, w'_{ac2}, w'_{ac}, R'_0, R'_{init}, R'_{init1}, R'_{11}, \ldots, R'_{1k_1}, R'_{ac1}, R'_{init2}, R'_{21}, \ldots, R'_{2k_2}, R'_{ac2}, R'_{ac})$$

ここに,

1. $\mu = [_0[_{init}]_{init}[_{init1}]_{init1}[_{11}]_{11} \cdots [_{1k_1}]_{1k_1}[_{ac1}]_{ac1} [_{init2}]_{init2}[_{21}]_{21} \cdots [_{2k_2}]_{2k_2}[_{ac2}]_{ac2}[_{ac}]_{ac}]_0$
   $[_{init}]_{init}$: initial membrane (初期薄膜)
   $[_{ac}]_{ac}$: accepting membrane (受理薄膜)

2. $w'_0 = \epsilon, w'_{init} = cq_0, w'_{ac} = q_f, w'_{init1} = q_{01}$, $w'_{ac1} = w_{ac1} = q_{f1}, w'_{init2} = q_{02}$, $w'_{ac2} = w_{ac2} = q_{f2}, w'_l = w_l$ $(l \in \{11, \ldots, 1k_1, 21, \ldots, 2k_2\})$,

3. (1) $R'_0 = R_{01} \cup R_{02} \cup R_1 \cup R_2 \cup R_3$,
   − $R_1 = \{cq_0 \to c_{in,init1}\, q_{0\,in,init}, cq_0 \to c_{in,init2}\, q_{0\,in,init}\}$,
   − $R_2 = \{cq_{f1} \to c_{in,ac}\, q_{f1\,in,ac1}\}$,
   − $R_3 = \{cq_{f2} \to c_{in,ac}\, q_{f2\,in,ac2}\}$,
   (2) $R'_{init} = \{cq_0 \to c_{out}\, q_{0\,out}\}$,
   (3) $R'_l = R_l$ $(l \in \{init1, 11, \ldots, 1k_1\})$,
   (4) $R'_l = R_l$ $(l \in \{init2, 21, \ldots, 2k_2\})$,
   (5) $R'_{ac1} = \{cq_{f1} \to c_{out}\, q_{f1\,out}\}$,
   (6) $R'_{ac2} = \{cq_{f2} \to c_{out}\, q_{f2\,out}\}$,
   (7) $R'_{ac} = \phi$.

初めに, G は 3.(2) の進化ルールを適用することにより, initial membrane $[_{init}]_{init}$ からオブジェクト cq_0 を skin membrane $[_0]_0$ に追い出す. その後, 3.(1) の R_1 の進化ルールを適用し, $cursor$ オブジェクト c を G_1 の initial membrane に相当する membrane $[_{init1}]_{init1}$ か G_2 の initial membrane に相当する membrane $[_{init2}]_{init2}$ に非決定的に追い出す. もし G が G_1 の計算を模倣する方を選択したとすると, 3.(3) の進化ルールが適用される. そして, $cursor$ オブジェクト c を G_1 の accepting membrane に相当する membrane $[_{ac1}]_{ac1}$ まで追い出し, G_1 に相当する入力文字列が受理されるのを模倣する. そして, G は 3.(5) の進化ルールを適用することにより, オブジェクト q_{f1} を membrane $[_{ac1}]_{ac1}$ から skin membrane に追い出す. そして, G は 3.(1) の R_2 の進化ルールを適用し, $cursor$ オブジェクト c を skin membrane から accepting membrane $[_{ac}]_{ac}$ に追い出し, 停止する. G が G_2 の計算を模倣する方を選択した場合も同様に動作する. つまり, $L(G) = L(E_1) \cup L(E_2)$ は明らかである.

Case I-2　$E = E_1E_2$ のとき : Case I-1 と同様にして $L(G) = L(E_1)L(E_2)$ を受理するような境界記号のない 1 方向 1NCPST(2) G を設計する. 詳細については省略する.

Case I-3　$E = {E_1}^*$ のとき : G_1 は Case I-1 で定義したものと同じとする. 境界記号のない 1 方向 1NCPST(2) G を次のように定義する.

$$G = (\{c\} \cup \{q_f\} \cup V_1, \Sigma_1, \mu, w'_0, w'_{init1}, w'_{11}, \ldots, w'_{1k_1}, w'_{ac1}, w'_{ac}, R'_0, R'_{init1}, R'_{11}, \ldots, R'_{1k_1}, R'_{ac1}, R'_{ac})$$

ここに,

1. $\mu = [_0[_{init1}]_{init1}[_{11}]_{11} \cdots [_{1k_1}]_{1k_1}[_{ac1}]_{ac1}[_{ac}]_{ac}]_0$
   $[_{init1}]_{init1}$: initial membrane (初期薄膜)
   $[_{ac}]_{ac}$: accepting membrane (受理薄膜)

2. $w'_0 = \epsilon, w'_{init1} = cq_{01}, w'_{ac1} = w_{ac1} = q_{f1}$, $w'_l = w_l$ $(l \in \{11, \ldots, 1k_1\}), w'_{ac} = w_{ac} = q_f$,

3. (1) $R'_0 = R_{01} \cup \cup R_1 \cup R_2 \cup R_3$,
   − $R_1 = \{cq_{01} \to c_{in,ac}\, q_{01\,in,init1}\}$,
   − $R_2 = \{cq_{f1} \to c_{in,ac}\, q_{f1\,in,ac1}\}$,
   − $R_3 = \{cq_{f1} \to c_{in,init1}\, q_{f1\,in,ac1}\}$,
   (2) $R'_l = R_l$ $(l \in \{init1, 11, \ldots, 1k_1\})$,
   (3) $R'_{ac1} = \{cq_{f1} \to c_{out}\, q_{f1\,out}\}$,
   (4) $R'_{ac} = \phi$.

初めに, G は 3.(2) の進化ルールを適用することにより, initial membrane $[_{init1}]_{init1}$ からオブジェ

クト cq_{01} を skin membrane $[_0]_0$ に追い出す．そして, 入力記号列 w が ϵ かそうでないかを非決定的に推測する．もし G が入力記号列 w を ϵ であると推測した場合は, 3.(1) の R_1 の進化ルールを適用することになり, 触手を右に動かすことなしに停止する．もし G が入力記号列 w を ϵ でないと推測した場合は, 入力記号列 w の初めの部分の w_1 の G_1 の計算の模倣を開始する．ここで, $w = w_1 \dots w_n$ $(n \geq 1)$ とする．もし G が $cursor$ オブジェクト c を G_1 の accepting membrane に相当する $[_{ac1}]_{ac1}$ まで追い出して, G_1 を受理した状態まで計算したとすると, 3.(3) の進化ルールを適用し, オブジェクト cq_{f1} を skin membrane に追い出す．そして, G は w_1 が w の最後の部分であるかどうか非決定的に推測する．もし G が w_1 が w の最後の部分だと推測した場合は, 3.(1) の R_2 の進化ルールを適用し, $cursor$ オブジェクト c を accepting membrane $[_{ac}]_{ac}$ に追い出し, 停止する．もし G が w_1 が w の最後の部分でないと推測した場合は, 3.(1) の R_3 の進化ルールが適用されて, G は $cursor$ オブジェクト c を initial membrane $[_{init1}]_{init1}$ に追い出し, w の 2 回目の部分の w_2 の G_1 の計算の模倣を開始する．以後, 停止するまで繰り返される．よって, $L(G) = L({E_1}^*)$ は明らかである．

以上により, 正則表現 E から直接, $L(E)$ を受理する境界記号のない 1 方向 1NCPST(2) を構成することができることが示された． $\square$

定理 4.1 より次の系を得る．

系 4.1 境界記号のない 1 方向 1NCPST(2) に対し,

$$\pounds[\text{1NCPST}(2)] = \text{REG}.$$

5 まとめと今後の課題

本研究では以下の結果が示せた．

- 各 $r \geq 1$, $X \in \{\text{D}, \text{N}\}$ に対して, $\pounds[1X\text{CPST}(r)] \subseteq \pounds[X\text{FA}]$.
- 各 $X \in \{\text{D}, \text{N}\}$ に対して, $\pounds[X\text{FA}] \subseteq \pounds[1X\text{CPST}(2)]$.
- 各 $r \geq 2$, $X \in \{\text{D}, \text{N}\}$ に対して, $\pounds[1X\text{CPST}(r)] = \pounds[1X\text{CPST}(2)] = \pounds[X\text{FA}]$.
- 各 $r \geq 2$ に対して, $\pounds[\text{1DCPST}(r)] = \pounds[\text{1NCPST}(r)]$.
- $\pounds[\text{DFA}] - \pounds[\text{1DCPST}(1)] \neq \emptyset$.
- $\pounds[\text{1DCPST}(1)] \not\subseteq \pounds[\text{1DCPST}(2)]$.
- 正則表現 E から直接（有限オートマトンを介さないで）, $L(E)$ を受理する境界記号のない 1 方向 1NCPST(2) を構成することができる．
- 境界記号のない 1 方向 1NCPST(2) に対して, $\pounds[\text{1NCPST}(2)] = \text{REG}$.

今後の課題として以下のことが挙げられる．

- 非決定性 1CPST から直接的に等価な決定性 1CPST を構成する方法を検討する．
- オブジェクトの個数を制限することによる受理能力の階層性が存在するかを調べる．
- membrane の個数を制限することによる受理能力の階層性が存在するかを調べる．

参考文献

[1] Gheorghe Paun, Grzegorz Rozenberg, A guid to membrane computing, Theoretical Computer Science 287 (2002),73-100

[2] Gh.Paun, Membrane Computing : An Introduction, Splinger-Verlag, 2002

[3] O.H.Ibarra, On the computational complexity of membrane computing systems, sudmitted, 2003

[4] O.H.Ibarra, The Number of Membranes Matters, manuscripts, Preproceedings of the Workshop on Membrame Computing(A.Alhazov et al.),Tarragona, July 17-22, 2003, 273-285

[5] Gh,Paun, List of problems circulated before the Brainstorming Week in Membrane Computing held in Tarragona, Spain, February 5-11, 2003

[6] 富田悦次, 横森貴, オートマトン・言語理論, 森北出版株式会社 (2002)