A note on parallel communicating
T0L array grammar systems

S. JEYA BHARATHI

Department of Mathematics, Thiagarajar College of Engineering, Madurai - 625 015 (India).

(Received: February 02, 2007; Accepted: May 22, 2007)

ABSTRACT

Parallel communicating L systems are systems having the communicating components as
L-systems. In parallel communicating L-array systems we have 0L-(T0L) Array systems as
communicating components. These models generates interesting pictures, some results on
generative capacity are given.

Keywords: PC0L array systems, PCT0L-array systems.

INTRODUCTION

Parallelism is a very modern topic in Computer Science, important from the practical
point of view and appealing from the theoretical one (Paun, 1992). L-Systems, introduced by
Lindenmayer in connection with some problems in theoretical Biology, involves parallelism at every level
of rewriting of the current sentential form (Rozenberg and Salomaa, 1980). Another type of
parallelism occurs in the framework of grammar systems. There one does not have one typical
generative device for producing a language, as in the classical formal language theory, but a system
of generative devices, working together in some
defined way for producing a language (Rozenberg and Paun). When the components of the system
work sequentially, cooperating/distributed grammar systems exists (Paun, 1992). This topic was
investigated in a series of papers (Csuhaj and Dassow 1993, Dassow and Paun 1990a, Dassow and Paun, 1990b, Paun 1992, Dassow and Paun 1996 from various points of view in Computer
Science: artificial intelligence, and psychology etc. with the motivation that the components of a
grammar system can also work in parallel and communicate in some way with each other. Thus the Parallel communicating grammar systems originate (Paun, 1992). In Parallel Communicating
(PC) grammar system, the components of the grammar system work in parallel (each having its
own sentential form) and communicate in some way (i.e when the query occurs, sending the currently
generated string to other component). This Parallel Communicating grammar system whose
components are L-systems (of various type), is mainly motivated by theoretical reasons, but seems
to have biological applications in symbiosis, parasitism, simultaneous interrelated growing of
plants and animals etc. Motivated by this we have proposed Parallel Communicating L-Array systems
and Parallel Communicating Tabled L-Array system. We consider a specific type of L-Array systems
defined in (Nirmal and Krithivasan, 1981) as the components of the system. This model is capable
of generating interesting pictures. We give some hierarchy results in communicating array grammar
and also examine some generative capacity.

Definition, examples and hierarchy
In this section we define PC0LAS (Parallel Communicating 0L Array System), PCT0LAS
(Parallel Communicating Tabled 0L Array System). We assume the reader to be familiar with the theory of 2D languages. We illustrate the model with examples. We present some more results on hierarchy.

**Definition**

Parallel Communicating 0L Array System (PC0LAS) is a 3-tuple \( G = (\Sigma, \omega, P) \) where 1. \( \Sigma = \Sigma' \cup K \), \( \Sigma' \) is basic alphabet, \( K = \{Q_1, Q_2, Q_3, \ldots, Q_n\} \) are query symbols. 2. \( \omega = \{\omega_1, \omega_2, \ldots, \omega_n\} \) is the axiom of \( n \)-tuple and \( w_i \in \Sigma^{**}, 1 \leq i \leq n \). 3. \( P = (P'_1, P'_2, P'_3, \ldots, P'_n) \) where \( P'_{ij} = P'_{ij} \cup P_{ij} \). Each \( P'_{ij} \) is a nonempty finite subset of \( \Sigma X \Sigma^{**} \), is a finite set of pairs \((a, x)\) with \( a \) in \( \Sigma \) and \( x \) in \( \Sigma^{**} \) of r x s dimensions such that for each \( a \) in \( \Sigma \) at least one such pair is in \( P'_{ij} \), \( 1 \leq i, j \leq n \). The pairs \((a, x)\) are called the rules or productions and are written as \( a \rightarrow x \). The communicating rules in \( P'_{ij} \) are of the form \( \alpha \rightarrow \beta \) \( \Delta \gamma \rightarrow \Delta \xi \), \( \alpha, \beta, \gamma, \delta, \xi \in \Sigma^{**} \), \( \Delta \) is the query symbol, and \( C \in \Sigma \).

The sets \( P''_{ij} \) maybe empty, for \( 1 \leq i, j \leq n \). The derivation step is of two types as

i. \( x_i \rightarrow y_i \) if \( x_i \) has no query symbols \( 1 \leq i \leq n, P'_{ij} \) contains \( y_i \), \( 1 \leq i, j \leq n \). \( x_i \) contains query symbols and then a communication step is performed, as these symbols impose. Specifically each symbol \( Q_i \) is replaced by the corresponding component \( G_i \), whereas the component \( G_i \), resumes working from its axiom. The communication has priority over rewriting. The Language generated by \( g \) is \( L(\gamma) = \{x \in (\Sigma - K)^++/ (\omega_1, \omega_2, \ldots, \omega_n) \rightarrow^* (x, \omega_1, \omega_2, \ldots, \omega_n), \omega_i \in \Sigma^{**}, 2 \leq i \leq n\} \).

In other words, a derivation consists of repeated rewriting and communication steps, starting from the \( n \)-tuple of axioms. We retain in \( L(\gamma) \) the array generated in this way on the first component, \( G_i \) (Which is considered the master of the system) without containing query symbols. We shall denote by \( PC(X) \) the family of languages generated by \( PC \) grammar having at most \( n \) components, \( n \geq 1 \).

i.e. To derive the configuration \((y_1, y_2, y_3, \ldots, y_n)\) either

- No query symbol appears in \( x_i, x_2, x_3, \ldots, x_n \) and then we have a component wise derivation, \( x_i = y_i, 1 \leq i \leq n \), then we have a component wise derivation, \( x_i = y_i, 1 \leq i \leq n \).

or

- Query symbols occur in some \( x_i \). Then a communication step is performed. Each occurrence of \( Q_j \) is replaced by \( x_j \) provided \( x_j \) does not contain symbols. In a communication step, the communicated string \( x_j \) replaces the query symbol \( Q_j \). Then the grammar \( G_j \) resumes rewriting beginning again from its axiom. The communication has priority over the effective rewriting: no rewriting is possible as long as at least one query symbol is present. If some query symbols are not satisfied at a given communication step, then they may be satisfied at the next step.

**Definition**

Parallel Communicating Tabled 0L Array System (PC0LAS) is a 3-tuple \( G = (\Sigma, \omega, P) \) where \( \Sigma, \omega \) are as in definition of PC0LAS. \( P = (P''_1, P''_2, P''_3, \ldots, P''_n) \) where \( P''_{ij} = P''_{ij} \cup P_{ij} \). Each \( P''_{ij} \) consists of a finite set \( \{P''_{ij} = P''_{ij} \cup P_{ij} \} \), \( f \geq 1 \). Each \( P''_{ij} \) is a finite subset of \( \Sigma X \Sigma^{**} \). called a table with the following two conditions.

i. \( (\forall a)_j (\exists a)_j \rightarrow (\exists a)_j, \langle a, a \rangle \in P\);  

ii. \( (\forall a)_j (\exists a)_j \rightarrow (\exists a)_j, a \) 's are of the same dimension.

The derivation step is of two types as

i. \( x_i \rightarrow y_i \) if \( x_i \) has no query symbols \( 1 \leq i \leq n \).  

ii. \( (x_1, x_2, x_3, \ldots, x_n) \rightarrow (y_1, y_2, y_3, \ldots, y_n) \) where some \( x_i \) has query symbols, \( Q_j \) and communicates with \( j \)-th component for \( 1 \leq i, j \leq n \).

The derivations are defined as follows \( P = P'' = \{P''_{ij} \cup P_{ij}\} \) where \( P'' = P''_{ij} \cup P_{ij} \) then we apply to \( \gamma \) the rules from the tables of \( P(\gamma) \) row by row (column by column) i.e., we choose a rule in \( P \) and apply the rules to the first row (column). Then we choose another set of rules in \( P \) and apply the rules
to the second row (column). Proceeding in this manner we apply the rules from a table to row (column) of $\omega$. If the query symbol appears, the communication takes place so as to get a rectangular array and if no query symbol appears in the master system then the communication does not work and the derivation stops. If the resultant array is rectangular then the rules from the table $P_{r}$ ($P_{c}$) can be applied again, otherwise the derivation comes to an end.

In PCTO$L$ array system, the query symbols can occur anywhere by rule provided. The resultant after the communication step, is rectangular. The row or column rules are applied in such a way that the rectangular format is maintained. The following figures are the models to form the rectangle pattern after the communication.

![Diagram](image)

2.3 Definition
Partial Parallel Communicating Tabled 0L Array System (Part CPCTO$\text{LAS}$) is a five tuple $G = (V, P, \omega, S, h)$ where

i) $(V, S, \omega)$ is a part PC0$\text{LAS}$

ii) $\Sigma$ is a non-empty finite set called target symbol.

iii) $h$ is a partial coding from $V$ into $\Sigma$.

Starting from $\omega$, arrays are derived by part PC$\text{TMLS}$ and then coding $h$ is applied to these arrays

$$h(a) = h(b) = h(c) = x$$

2.3 Definition
Partial Parallel Communicating Tabled 0L Array System (Part CPCTO$\text{LAS}$) is a five tuple $G = (V, P, \omega, S, h)$ where

i) $(V, S, \omega)$ is a part PC0$\text{LAS}$

ii) $\Sigma$ is a non-empty finite set called target symbol.

iii) $h$ is a partial coding from $V$ into $\Sigma$.

Starting from $\omega$, arrays are derived by part PC$\text{TMLS}$ and then coding $h$ is applied to these arrays.

$$h(a) = h(b) = h(c) = x$$

Model
Let $G = (V, P, \omega, S, h)$ be a part CPCTAS (of degree four) $\gamma = (G_{1}, G_{2}, G_{3}, G_{4})$.

where $V = (X, Y, Z, a, b, c, Q_{2}, Q_{3}, Q_{4})$

$\sum = (a, b, c)$

Starting from $\omega$, arrays are derived by part PC$\text{TMLS}$ and then coding $h$ is applied to these arrays.

$$h(a) = h(b) = h(c) = x$$

$$\omega' = \begin{cases}
X \\
a, b, Y, c, X, b \\
Z
\end{cases}$$

$$\Rightarrow \begin{cases}
X \\
b, b, b, Y, c, X, b, b, b \\
a, c \\
c, z \\
Q_{2}, X \\
b, b, Q_{3}, b, b, b, b, b \\
c, c, c, z \\
Q_{4}
\end{cases}$$
Clearly $X = PC_1X \subseteq PC_2X \subseteq \ldots$ for all $X \in \{0LAL, T0LAL\}$. We shall show that $X \subset PC_2X$ (two components are strictly powerful than one) for $X \in \{0LAL, T0LAL\}$.

**Theorem**

(i) $PC_{0LAL} \subsetneq E0LAL$.

(ii) $PC_{0LAL} \subsetneq 0LAL$.

**Proof**: Follows from the definition of $PC_{0LAL}$. In particular, model-1 cannot be generated by any $E0LAL$.

**Corollary**

(i) $0LAL \subset PC_2 0LAL$.

(ii) $T0LAL \subset PC_2 T0LAL$.

**Theorem**

The language $L = \{ \{a\}, \{a\} \}$ is not in $PC_0T0LAL, n \geq 1$.

**Proof**:

Suppose $L = L(\gamma)$ for a $PC_0 T0LAL$ where $\gamma = (G_1, G_2, G_3, \ldots, G_n)$ be the $T0LAL$ components, $G_i = (V_i, \omega_i, \{P_{i1}, P_{i2}, \ldots, P_{in}\})$, $1 \leq i \leq n$. $V_i = \{a\}$.
generated by PCT0LAS array system (of degree two) \( \gamma = (G_1, G_2) \)

let \( G = (\{ a, b, c, Q_2 \}, (a, c), P) \)

\[ P = P_1 \cup P_2 \]

\[
P_1 = \begin{cases} 
  c & b & a & c & c & c & c \\
  c & b & a & b & c & c & c \\
  c & b & a & a & b & a & c \\
  \# & \# & c & c & \# & \# & c \\
\end{cases}
\]

\[
P_2 = \begin{cases} 
  a & a \\
  c & a & a \\
  \end{cases}
\]

\[
\gamma = \{ a, a \}
\]

This will generate the language

\[
\gamma L = \{ a, a \}
\]

So arrays, which are not in \( L \), will be generated by \( G \). Hence \( L \) is not a PC0LAL.

**Theorem**

There exists an algorithm which for an arbitrary PCTAS \( G \) produces \( G' \) and a partial coding \( h \) such that \( L(G) = h(L(G')) \) in the normal form

**Proof:**

Let \( G = (V, P, \omega, \Sigma) \) be an PCTAS, then

Let \( G' = (V \cup (S, F), P', h) \) be an part CPCTAS, where \( (S, F) \notin V \) and \( P' = P_0 \cup (P_i \forall P_i \in P) \)

Let \( h \) be a partial coding \( h: V \rightarrow \Sigma' \), such that \( h \) is defined if \( h(a) \) defined for all \( i,j \) \( h \) is not defined if atleast one \( a_{ij} \) is not defined.

The table in \( P' \) was constructed in such a way as to provide a possibility for the following simulation (in \( G' \)) of derivations in \( G \).

Let \( P_0 = (S \rightarrow \omega, a \rightarrow M'r_{s_0} ; a \in V \cup (F) \) )

\[
P' = P_0 \cup (S \rightarrow M'r_{s_i}, F \rightarrow M'r_{s_i}) \text{ for } 1 \leq i \leq n
\]

Where \( M'r_{s_i} \) are rejection arrays.

Hence it is clear that \( L(G) = L(G') \).

If \( D \) is a derivation in \( G \), then we construct
a corresponding derivation $D^o$ in $G^o$ in such a way that $h$ gives the partial coding. We keep track of all productive occurrences such that the partial coding $h$ is defined for all.

**Conclusion**

A new type of communicating L-array model is formulated in this paper. This model generates interesting pictures. Some results on hierarchy are given. Seven other variation of this model can be made using the concept of E0LAL and ET0LAL.

**REFERENCES**

1. CsuhaJ-Varju E, Dassow J, On Cooperating/distributed grammar systems, Elektr. Inform Kybern, (1993).
2. Dassow J., Paun Gh, Regulatedrewriting in formal languages theory, Akademic Verlag, Berlin, Springer-Verlag, Berlin, 1990a (1989).
3. Dassow J., Paun Gh: On some variants of cooperating/distributed grammar systems, Stud. Cerc. Matem, 42(2), 153-165 (1990b).
4. Gheoreghe Paun: Parallel Communicating Grammar Systems of L-systems, Springer-Verlag, 405-417 (1992).
5. N. Nirmal & K. Krithivasan, E0L and ET0L Array Languages, Proc. Indian AcadSci (Math. Science), 90(3), 167-180 (1981).
6. G. Rozenberg and A. Salomaa, The Mathematical Theory of L-Systems, AcademicPress, New York, (1980).