January – February 2020
ISSN: 0193-4120 Page No. 13304 - 13309

Certain Comparison Results on Parallel Contextual Array Insertion Deletion Grammar

T. Kamaraj[1], S. Jayasankar[2], James Immanuel[3], D.G. Thomas[4]

[1]Department of Mathematics, Sathyabama Institute of Science and Technology, Tamilnadu, Chennai 600119, India
[2] Ramakrishna Mission Vivekananda College, Mylapore, Chennai 600004, Tamilnadu, India
[3]Department of Science and Humanities (Mathematics Division), Saveetha School of Engineering, Tamilnadu, Chennai 602105, India
[1]kamarajmx@gmail.com, [2]ksjayjay@gmail.com, [3]james_imch@yahoo.co.in,[4]dgthomasmcc@yahoo.com

Article Info
Volume 82
Page Number: 13304 - 13309
Publication Issue:
January-February 2020

Article History
***Article Received**: 18 May 2019*
***Revised**: 14 July 2019*
***Accepted**: 22 December 2019*
***Publication**: 24 February 2020*

Abstract

Parallel Contextual array insertion and deletion grammars (PCAIDG) are the recently introduced formalism for generating rectangular array languages using insertion and deletion operations and parallel contextual mappings. PCAIDG strictly include Tiling System (TS) of Giammaresi and Restivo. On the other hand, a context free class of grammar model named, Tile rewriting grammars (TRG) is also introduced in recent time for rectangular picture generation by amalgamating isometric rewriting rules with the Tiling System. TRG is proved to be more powerful than TS and array grammars by Siromoney and Prusa. In this work, we compare PCAIDG with TRG to understand the expressiveness in depth and form a hierarchy of these classes of grammars.

Keywords; *Parallel contextual array - tiling system – tiling rewriting grammar*

I. INTRODUCTION

Tiling system (TS) of Giammaresi and Restivo [1] is a concept of recognizability of a set of pictures belonging to the family REC, defined by projection of local languages. Tile rewriting grammars (TRG) and its variant, regional tile rewriting grammars (RTG), are the recent context free classes of grammars [4] which combined isometric rewriting rules with the tiling system. TRG is proved to be more powerful than TS and array grammars by Siromoney and RTG.

A new array contextual grammar called parallel contextual array insertion deletion grammar (PCAIDG) has been introduced in [2], based on internal parallel contextual grammars. In this paper, we compare the generative power of parallel contextual array insertion deletion grammar (PCAIDG) with tile rewriting grammar (TRG) and regional tile rewriting grammar (RTG).

II. PRELIMINARIES

For the definitions pertaining to tile rewriting grammar (TRG) we refer to [4] and for parallel contextual array insertion deletion grammar we refer to [2].

III. RESULTS

In this section, we compare the generative power of PCAIDG with TRG and RTG. We denote by $\mathcal{L}(G)$, the family of languages generated by the grammar G. The family of all array languages generated by parallel contextual array insertion deletion grammars is denoted by $\mathcal{L}(PCAIDG)$.

Theorem 1.$\boldsymbol{\mathcal{L}(PCAIDG) \cap \mathcal{L}(TRG) \neq \emptyset}$.

$Proof$. The language L_1 consisting of pictures with palindromic columns of ①$'s$ and ②$'s$ with even number of rows i.e.,

Published by: The Mattingley Publishing Co., Inc.

January – February 2020
ISSN: 0193-4120 Page No. 13304 - 13309

$$L_1=\left\{\begin{matrix}\textcircled{1} & \textcircled{1}\\ \textcircled{1} & \textcircled{1}\end{matrix},\ \begin{matrix}\textcircled{1} & \textcircled{2}\\ \textcircled{1} & \textcircled{2}\end{matrix},\ \begin{matrix}\textcircled{2} & \textcircled{1}\\ \textcircled{2} & \textcircled{1}\end{matrix},\ \begin{matrix}\textcircled{2} & \textcircled{2}\\ \textcircled{2} & \textcircled{2}\end{matrix},\ \begin{matrix}\textcircled{1} & \textcircled{1} & \textcircled{1}\\ \textcircled{1} & \textcircled{1} & \textcircled{1}\\ \textcircled{1} & \textcircled{1} & \textcircled{1}\\ \textcircled{1} & \textcircled{1} & \textcircled{1}\end{matrix},\ \begin{matrix}\textcircled{1} & \textcircled{2} & \textcircled{2}\\ \textcircled{2} & \textcircled{2} & \textcircled{2}\\ \textcircled{2} & \textcircled{2} & \textcircled{2}\\ \textcircled{1} & \textcircled{2} & \textcircled{2}\end{matrix},\ \dots\right\}$$

is proved to be in $\mathcal{L}(TRG)$[4].

We construct a PCAIDG grammar G to generate L_1.

Let $G=\left(V,T,M,C,R,\varphi_r^i,\varphi_c^i,\varphi_r^d,\varphi_c^d\right)$ with $V=T\cup\{\blacksquare,\square\}, T=\{\textcircled{1},\textcircled{2}\}$.

$$M=\left\{\begin{bmatrix}\blacksquare & \blacksquare\\ \blacksquare & \textcircled{1}\end{bmatrix},\begin{bmatrix}\blacksquare & \blacksquare\\ \blacksquare & \textcircled{2}\end{bmatrix}\right\}, C=\left\{\begin{bmatrix}\blacksquare\\ \textcircled{1}\end{bmatrix},\begin{bmatrix}\blacksquare\\ \textcircled{2}\end{bmatrix},\begin{bmatrix}\blacksquare\\ \blacksquare\end{bmatrix}\right\},$$

$$R=\left\{\begin{bmatrix}\blacksquare & \square\\ \blacksquare & \square\\ \blacksquare & \textcircled{1}\\ \blacksquare & \blacksquare\end{bmatrix},\begin{bmatrix}\blacksquare & \square\\ \blacksquare & \square\\ \blacksquare & \textcircled{2}\\ \blacksquare & \blacksquare\end{bmatrix},\begin{bmatrix}\square & \square\\ \square & \square\\ \textcircled{1} & \textcircled{2}\\ \blacksquare & \blacksquare\end{bmatrix},\begin{bmatrix}\square & \square\\ \square & \square\\ \textcircled{1} & \textcircled{1}\\ \blacksquare & \blacksquare\end{bmatrix},\begin{bmatrix}\square & \square\\ \square & \square\\ \textcircled{2} & \textcircled{1}\\ \blacksquare & \blacksquare\end{bmatrix},\begin{bmatrix}\square & \square\\ \square & \square\\ \textcircled{2} & \textcircled{2}\\ \blacksquare & \blacksquare\end{bmatrix}\right\}.$$

For $\alpha,\beta,\gamma\in\{\textcircled{1},\textcircled{2}\}$, we define column insertion mappings as follows:

$$\varphi_c^i\begin{bmatrix}\blacksquare & \blacksquare, & \lambda\\ \blacksquare & \alpha, & \lambda\end{bmatrix}=\begin{Bmatrix}\blacksquare\\ \beta\end{Bmatrix},\ \varphi_c^i\begin{bmatrix}\blacksquare & \blacksquare, & \lambda\\ \blacksquare & \alpha, & \lambda\end{bmatrix}=\begin{Bmatrix}\blacksquare\\ \blacksquare\end{Bmatrix},\varphi_c^i\begin{bmatrix}\blacksquare & \blacksquare, & \lambda\\ \alpha & \beta, & \lambda\end{bmatrix}=\begin{Bmatrix}\blacksquare & \blacksquare\\ \gamma, & \blacksquare\end{Bmatrix}$$

For $\alpha,\beta,u,v\in\{\textcircled{1},\textcircled{2}\}$, we define row insertion mappings as follows:

$$\varphi_r^i\begin{bmatrix}\blacksquare & \blacksquare, & \lambda & \lambda\\ \blacksquare & \alpha, & & \end{bmatrix}=\begin{Bmatrix}\blacksquare & \square\\ \blacksquare & \square\\ \blacksquare & \alpha\\ \blacksquare & \blacksquare\end{Bmatrix},\varphi_r^i\begin{bmatrix}\blacksquare & \blacksquare, & \lambda & \lambda\\ \alpha & \beta, & & \end{bmatrix}=\begin{Bmatrix}\square & \square\\ \square & \square\\ \alpha & \beta\\ \blacksquare & \blacksquare\end{Bmatrix},$$

$$\varphi_r^i\begin{bmatrix}\blacksquare & \blacksquare, & \lambda & \lambda\\ \alpha & \blacksquare, & & \end{bmatrix}=\begin{Bmatrix}\square & \blacksquare\\ \square & \blacksquare\\ \alpha & \blacksquare\\ \blacksquare & \blacksquare\end{Bmatrix},\varphi_r^i\begin{bmatrix}\blacksquare & \alpha, & \blacksquare & \square\\ \blacksquare & \square, & \blacksquare & \alpha\end{bmatrix}=\begin{Bmatrix}\blacksquare & \beta\\ \blacksquare & \square\\ \blacksquare & \square\\ \blacksquare & \beta\end{Bmatrix},$$

$$\varphi_r^i\begin{bmatrix}\alpha & \beta, & \square & \square\\ \square & \square, & \alpha & \beta\end{bmatrix}=\begin{Bmatrix}u & v\\ \square & \square\\ \square & \square\\ u & v\end{Bmatrix},\varphi_r^i\begin{bmatrix}\alpha & \blacksquare, & \square & \blacksquare\\ \square & \blacksquare, & \alpha & \blacksquare\end{bmatrix}=\begin{Bmatrix}\beta & \blacksquare\\ \square & \blacksquare\\ \square & \blacksquare\\ \beta & \blacksquare\end{Bmatrix}.$$

For $a,b,u,v\in\{\textcircled{1},\textcircled{2}\}$, we define deletion operations as follows:

$\varphi_r^d[\lambda\ \ \lambda,\ \ \blacksquare\ \ u]=\{\blacksquare\ \ \blacksquare\},\ \varphi_r^d[\lambda\ \ \lambda,\ \ u\ \ v]=\{\blacksquare\ \ \blacksquare\},$

$\varphi_r^d[\lambda\ \ \lambda,\ \ u\ \ \blacksquare]=\{\blacksquare\ \ \blacksquare\},\ \varphi_r^d[\blacksquare\ \ u,\ \ \lambda\ \ \lambda]=\{\blacksquare\ \ \blacksquare\},$

$\varphi_r^d[u\ \ v,\ \ \lambda\ \ \lambda]=\{\blacksquare\ \ \blacksquare\},\varphi_r^d[u\ \ \blacksquare,\ \ \lambda\ \ \lambda]=\{\blacksquare\ \ \blacksquare\},$

$\varphi_r^d[\blacksquare\ \ \alpha,\ \ \blacksquare\ \ \beta]=\{\blacksquare\ \ \square\},\ \varphi_r^d[\alpha\ \ \beta,\ \ u\ \ v]=\{\square\square\},$

$\varphi_r^d[\alpha\ \ \blacksquare,\ \ \beta\ \ \blacksquare]=\{\square\ \ \blacksquare\},\varphi_r^d[\blacksquare\ \ \alpha,\ \ \blacksquare\ \ \alpha]=\begin{Bmatrix}\blacksquare & \square\\ \blacksquare & \square\end{Bmatrix},$

January – February 2020
ISSN: 0193-4120 Page No. 13304 - 13309

$$\varphi_r^d[\alpha \quad \beta\,, \quad \alpha \quad \beta] = \left\{\begin{matrix} \text{▱} & \text{▱} \\ \text{▱} & \text{▱} \end{matrix}\right\},\ \varphi_r^d[\alpha \quad \text{▤}\,, \quad \alpha \quad \text{▤}] = \left\{\begin{matrix} \text{▱} & \text{▤} \\ \text{▱} & \text{▤} \end{matrix}\right\}.$$

$$\varphi_c^d\begin{bmatrix} \lambda, & \text{▤} \\ \lambda, & u \end{bmatrix} = \left\{\begin{matrix} \text{▤} \\ \text{▤} \end{matrix}\right\}, \varphi_c^d\begin{bmatrix} \lambda, & u \\ \lambda, & v \end{bmatrix} = \left\{\begin{matrix} \text{▤} \\ \text{▤} \end{matrix}\right\}, \varphi_c^d\begin{bmatrix} \lambda, & u \\ \lambda, & \text{▤} \end{bmatrix} = \left\{\begin{matrix} \text{▤} \\ \text{▤} \end{matrix}\right\},$$

$$\varphi_c^d\begin{bmatrix} \text{▤}, & \lambda \\ u, & \lambda \end{bmatrix} = \left\{\begin{matrix} \text{▤} \\ \text{▤} \end{matrix}\right\}, \qquad \varphi_c^d\begin{bmatrix} u, & \lambda \\ v, & \lambda \end{bmatrix} = \left\{\begin{matrix} \text{▤} \\ \text{▤} \end{matrix}\right\}, \qquad \varphi_c^d\begin{bmatrix} u, & \lambda \\ \text{▤}, & \lambda \end{bmatrix} = \left\{\begin{matrix} \text{▤} \\ \text{▤} \end{matrix}\right\}.$$

The working rule is as follows:

Starting with the axiom set, first the column insertion rules are applied for the horizontal growth of the picture and then vertical picture growth is achieved by the insertion of rows with ▱'s at the middle so as to preserve the palindromic structure. Now two possibilities arise: either the picture with terminal symbols can be obtained by applying deletion rules or further growth can be achieved by continuing with insertion rules maintaining palindromic structure.

$$M \to \begin{matrix} \text{▤} & \text{▤} \\ \text{▤} & \text{①} \end{matrix} \Rightarrow_i^{col} \begin{matrix} \text{▤} & \text{▤} & \text{▤} \\ \text{▤} & \text{①} & \text{②} \end{matrix} \Rightarrow_i^{col} \begin{matrix} \text{▤} & \text{▤} & \text{▤} & \text{▤} \\ \text{▤} & \text{①} & \text{②} & \text{②} \end{matrix} \Rightarrow_i^{col} \begin{matrix} \text{▤} & \text{▤} & \text{▤} & \text{▤} & \text{▤} \\ \text{▤} & \text{①} & \text{②} & \text{②} & \text{▤} \end{matrix} \Rightarrow_i^{row}$$

$$\begin{matrix} \text{▤} & \text{▤} & \text{▤} & \text{▤} & \text{▤} \\ \text{▤} & \text{①} & \text{②} & \text{②} & \text{▤} \\ \text{▤} & \text{▱} & \text{▱} & \text{▱} & \text{▤} \\ \text{▤} & \text{▱} & \text{▱} & \text{▱} & \text{▤} \\ \text{▤} & \text{①} & \text{②} & \text{②} & \text{▤} \\ \text{▤} & \text{▤} & \text{▤} & \text{▤} & \text{▤} \end{matrix} \Rightarrow_i^{row} \begin{matrix} \text{▤} & \text{▤} & \text{▤} & \text{▤} & \text{▤} \\ \text{▤} & \text{①} & \text{②} & \text{②} & \text{▤} \\ \text{▤} & \text{▱} & \text{▱} & \text{▱} & \text{▤} \\ \text{▤} & \text{②} & \text{①} & \text{②} & \text{▤} \\ \text{▤} & \text{②} & \text{①} & \text{②} & \text{▤} \\ \text{▤} & \text{▱} & \text{▱} & \text{▱} & \text{▤} \\ \text{▤} & \text{①} & \text{②} & \text{②} & \text{▤} \\ \text{▤} & \text{▤} & \text{▤} & \text{▤} & \text{▤} \end{matrix} \Rightarrow_i^{row} \begin{matrix} \text{▤} & \text{▤} & \text{▤} & \text{▤} & \text{▤} \\ \text{▤} & \text{①} & \text{②} & \text{②} & \text{▤} \\ \text{▤} & \text{▱} & \text{▱} & \text{▱} & \text{▤} \\ \text{▤} & \text{②} & \text{①} & \text{②} & \text{▤} \\ \text{▤} & \text{▱} & \text{▱} & \text{▱} & \text{▤} \\ \text{▤} & \text{▱} & \text{▱} & \text{▱} & \text{▤} \\ \text{▤} & \text{②} & \text{①} & \text{②} & \text{▤} \\ \text{▤} & \text{▱} & \text{▱} & \text{▱} & \text{▤} \\ \text{▤} & \text{①} & \text{②} & \text{②} & \text{▤} \\ \text{▤} & \text{▤} & \text{▤} & \text{▤} & \text{▤} \end{matrix} \Rightarrow_d^{row}$$

$$\begin{matrix} \text{▤} & \text{▤} & \text{▤} & \text{▤} & \text{▤} \\ \text{▤} & \text{①} & \text{②} & \text{②} & \text{▤} \\ \text{▤} & \text{②} & \text{①} & \text{②} & \text{▤} \\ \text{▤} & \text{▱} & \text{▱} & \text{▱} & \text{▤} \\ \text{▤} & \text{▱} & \text{▱} & \text{▱} & \text{▤} \\ \text{▤} & \text{②} & \text{①} & \text{②} & \text{▤} \\ \text{▤} & \text{▱} & \text{▱} & \text{▱} & \text{▤} \\ \text{▤} & \text{①} & \text{②} & \text{②} & \text{▤} \\ \text{▤} & \text{▤} & \text{▤} & \text{▤} & \text{▤} \end{matrix} \Rightarrow_d^{row} \begin{matrix} \text{▤} & \text{▤} & \text{▤} & \text{▤} & \text{▤} \\ \text{▤} & \text{①} & \text{②} & \text{②} & \text{▤} \\ \text{▤} & \text{②} & \text{①} & \text{②} & \text{▤} \\ \text{▤} & \text{▱} & \text{▱} & \text{▱} & \text{▤} \\ \text{▤} & \text{▱} & \text{▱} & \text{▱} & \text{▤} \\ \text{▤} & \text{②} & \text{①} & \text{②} & \text{▤} \\ \text{▤} & \text{①} & \text{②} & \text{②} & \text{▤} \\ \text{▤} & \text{▤} & \text{▤} & \text{▤} & \text{▤} \end{matrix} \Rightarrow_d^{row} \begin{matrix} \text{▤} & \text{▤} & \text{▤} & \text{▤} & \text{▤} \\ \text{▤} & \text{①} & \text{②} & \text{②} & \text{▤} \\ \text{▤} & \text{②} & \text{①} & \text{②} & \text{▤} \\ \text{▤} & \text{②} & \text{①} & \text{②} & \text{▤} \\ \text{▤} & \text{①} & \text{②} & \text{②} & \text{▤} \\ \text{▤} & \text{▤} & \text{▤} & \text{▤} & \text{▤} \end{matrix} \Rightarrow^{d}$$

$$\begin{matrix} \text{①} & \text{②} & \text{②} \\ \text{②} & \text{①} & \text{②} \\ \text{②} & \text{①} & \text{②} \\ \text{①} & \text{②} & \text{②} \end{matrix}.$$

Theorem 2. $\boldsymbol{\mathcal{L}(PCAIDG) \cap \mathcal{L}(RTG) \neq \emptyset}$.

Proof. The language L_2 consisting of pictures with perpendicular ②'s (not along borders) in the background of ①'s i.e.,

Published by: The Mattingley Publishing Co., Inc.

January – February 2020
ISSN: 0193-4120 Page No. 13304 - 13309

$$L_2=\left\{\begin{matrix}\textcircled{1}&\textcircled{2}&\textcircled{1}\\ \textcircled{2}&\textcircled{2}&\textcircled{2}\\ \textcircled{1}&\textcircled{2}&\textcircled{1}\end{matrix}\,,\ \begin{matrix}\textcircled{1}&\textcircled{2}&\textcircled{1}&\textcircled{1}\\ \textcircled{1}&\textcircled{2}&\textcircled{1}&\textcircled{1}\\ \textcircled{2}&\textcircled{2}&\textcircled{2}&\textcircled{2}\\ \textcircled{1}&\textcircled{2}&\textcircled{1}&\textcircled{1}\end{matrix}\,,\ \begin{matrix}\textcircled{1}&\textcircled{1}&\textcircled{2}&\textcircled{1}&\textcircled{1}\\ \textcircled{1}&\textcircled{1}&\textcircled{2}&\textcircled{1}&\textcircled{1}\\ \textcircled{2}&\textcircled{2}&\textcircled{2}&\textcircled{2}&\textcircled{2}\\ \textcircled{1}&\textcircled{1}&\textcircled{2}&\textcircled{1}&\textcircled{1}\\ \textcircled{1}&\textcircled{1}&\textcircled{2}&\textcircled{1}&\textcircled{1}\end{matrix}\,,\ \ldots\right\}$$ is proved to be in $\mathcal{L}(RTG)$ [4].

We now construct a PCAIDG grammar G to generate L_2.

Let $G=\left(V,T,M,C,R,\varphi_r^i,\varphi_c^i,\varphi_r^d,\varphi_c^d\right)$ with $V=T\cup\{\boxminus,\square,\boxdot\}$, $T=\{\textcircled{1},\textcircled{2}\}$.

$M=\left\{\begin{bmatrix}\boxminus&\boxminus&\boxminus&\boxminus\\ \boxminus&\textcircled{1}&\square&\boxminus\end{bmatrix}\right\}, C=\left\{\begin{bmatrix}\boxminus\\ \textcircled{1}\end{bmatrix},\begin{bmatrix}\boxminus\\ \textcircled{2}\end{bmatrix},\begin{bmatrix}\boxminus\\ \square\end{bmatrix},\begin{bmatrix}\boxminus\\ \boxminus\end{bmatrix}\right\}.$

We define column insertion mappings as follows:

$\varphi_c^i\begin{bmatrix}\boxminus&\boxminus\\ \textcircled{1},&\square\end{bmatrix}=\left\{\begin{matrix}\boxminus&\boxminus\\ \textcircled{1},&\textcircled{2}\end{matrix}\right\}, \varphi_c^i\begin{bmatrix}\boxminus&\boxminus&\boxminus\\ \textcircled{2}&\square,&\boxminus\end{bmatrix}=\left\{\begin{matrix}\boxminus\\ \textcircled{1}\end{matrix}\right\}, \varphi_c^i\begin{bmatrix}\boxminus&\boxminus\\ \textcircled{1},&\boxminus\end{bmatrix}=\left\{\begin{matrix}\boxminus\\ \textcircled{1}\end{matrix}\right\}.$

We now define row insertion mappings as follows:

$\phi_r^i\begin{bmatrix}\boxminus&\boxminus&&\\ \boxminus&\textcircled{1},&\lambda&\lambda\end{bmatrix}=\left\{\begin{matrix}\boxminus&\boxdot\\ \boxminus&\boxminus\end{matrix}\right\}, \phi_r^i\begin{bmatrix}\boxminus&\boxminus&&\\ \textcircled{1}&\textcircled{1},&\lambda&\lambda\end{bmatrix}=\left\{\begin{matrix}\boxdot&\boxdot\\ \boxminus&\boxminus\end{matrix}\right\}, \phi_r^i\begin{bmatrix}\boxminus&\boxminus&&\\ \textcircled{1}&\textcircled{2},&\lambda&\lambda\end{bmatrix}=\left\{\begin{matrix}\boxdot&\boxdot\\ \boxminus&\boxminus\end{matrix}\right\},$
$\phi_r^i\begin{bmatrix}\boxminus&\boxminus&&\\ \textcircled{2}&\textcircled{1},&\lambda&\lambda\end{bmatrix}=\left\{\begin{matrix}\boxdot&\boxdot\\ \boxminus&\boxminus\end{matrix}\right\},$

$\phi_r^i\begin{bmatrix}\boxminus&\boxminus&&\\ \textcircled{1}&\boxminus,&\lambda&\lambda\end{bmatrix}=\left\{\begin{matrix}\boxdot&\boxminus\\ \boxminus&\boxminus\end{matrix}\right\}, \phi_r^i[\boxminus\ \ \textcircled{1},\ \ \boxminus\ \ \boxdot]=\{\boxminus\ \ \textcircled{1},\ \ \boxminus\ \ \textcircled{2}\}$

$\phi_r^i[\textcircled{1}\ \ \textcircled{1},\ \ Z\ \ Z]=\{\textcircled{1}\ \ \textcircled{1},\ \ \textcircled{2}\ \ \textcircled{2}\},$

$\phi_r^i[\textcircled{1}\ \ \textcircled{2},\ \ \boxdot\ \ \boxdot]=\{\textcircled{1}\ \ \textcircled{2},\ \ \textcircled{2}\ \ \textcircled{2}\},$

$\phi_r^i[\textcircled{2}\ \ \textcircled{1},\ \ \boxdot\ \ \boxdot]=\{\textcircled{2}\ \ \textcircled{1},\ \ \textcircled{2}\ \ \textcircled{2}\},$

$\phi_r^i[\textcircled{1}\ \ \boxminus,\ \ \boxdot\ \ \boxminus]=\{\textcircled{1}\ \ \boxminus,\ \ \textcircled{2}\ \ \boxminus\}$

$\phi_r^i\begin{bmatrix}\boxminus&\textcircled{1}&&\\ \boxminus&\textcircled{2},&\boxminus&\boxminus\\ \boxminus&\boxdot&&\end{bmatrix}=\{\boxminus\ \ \textcircled{1}\}, \phi_r^i\begin{bmatrix}\textcircled{1}&\textcircled{1}&&\\ \textcircled{2}&\textcircled{2},&\boxminus&\boxminus\\ \boxdot&\boxdot&&\end{bmatrix}=\{\textcircled{1}\ \ \textcircled{1}\},$

$\phi_r^i\begin{bmatrix}\textcircled{1}&\boxminus&&\\ \textcircled{2}&\boxminus,&\boxminus&\boxminus\\ \boxdot&\boxminus&&\end{bmatrix}=\{\textcircled{1}\ \ \boxminus\}, \phi_r^i\begin{bmatrix}\textcircled{1}&\textcircled{2}&&\\ \textcircled{2}&\textcircled{2},&\boxminus&\boxminus\\ \boxdot&\boxdot&&\end{bmatrix}=\{\textcircled{1}\ \ \textcircled{2}\},$

$\phi_r^i\begin{bmatrix}\textcircled{2}&\textcircled{1}&&\\ \textcircled{2}&\textcircled{2},&\boxminus&\boxminus\\ \boxdot&\boxdot&&\end{bmatrix}=\{\textcircled{2}\ \ \textcircled{1}\}, \phi_r^i\begin{bmatrix}\boxminus&\textcircled{1}&&\\ \boxminus&\textcircled{1},&\boxminus&\boxminus\end{bmatrix}=\{\boxminus\ \ \textcircled{1}\}, \phi_r^i\begin{bmatrix}\textcircled{1}&\textcircled{1}&&\\ \textcircled{1}&\textcircled{1},&\boxminus&\boxminus\end{bmatrix}=\{\textcircled{1}\ \ \textcircled{1}\},$
$\phi_r^i\begin{bmatrix}\textcircled{1}&\boxminus&&\\ \textcircled{1}&\boxminus,&\boxminus&\boxminus\end{bmatrix}=\{\textcircled{1}\ \ \boxminus\},$

Published by: The Mattingley Publishing Co., Inc.

January – February 2020
ISSN: 0193-4120 Page No. 13304 - 13309

ϕ_r^i [① ② / ① ① , ▨ ▨] = {① ②}, ϕ_r^i [② ① / ① ① , ▨ ▨] = {② ①}.

We now define the deletion mappings as follows:

φ_r^d[▨ ① , ▨ ②] = {▨ ▣}, φ_r^d[① ② , ② ②] = {▣ ▣},

φ_r^d[② ① , ② ②] = {▣ ▣}, φ_r^d[① ① , ② ②] = {▣ ▣},

φ_r^d[x ▨ , ② ▨] = {▣ ▨}, φ_r^d[① ② , ▨ ▨] = {▨ ▨},

φ_r^d[② ① , ▨ ▨] = {▨ ▨}, φ_r^d[① ① , ▨ ▨] = {▨ ▨},

φ_r^d[λ λ , ① ②] = {▨ ▨}, φ_r^d[λ λ , ② ①] = {▨ ▨},

φ_r^d[λ λ , ① ①] = {▨ ▨}, φ_r^d[λ λ , ① ▨] = {▨ ▨},

φ_r^d[① ② , λ λ] = {▨ ▨}, φ_r^d[② ① , λ λ] = {▨ ▨},

φ_r^d[① ① , λ λ] = {▨ ▨}, φ_r^d[① ▨ , λ λ] = {① ▨},

φ_r^d[▨ ② , ▨ ①] = {▨ ▣}, φ_r^d[② ② , ① ①] = {▣ ▣},

φ_r^d[② ▨ , ① ▨] = {▣ ▨}.

φ_c^d [λ / λ , ▨ / ①] = {▨ / ▨}, φ_c^d [λ / λ , ① / ①] = {▨ / ▨}, φ_c^d [λ / λ , ① / ②] = {▨ / ▨},

φ_c^d [λ / λ , ② / ①] = {▨ / ▨}, φ_c^d [λ / λ , ① / ▨] = {▨ / ▨}, φ_c^d [▨ / ① , λ / λ] = {▨ / ▨},

φ_c^d [① / ② , λ / λ] = {▨ / ▨}, φ_c^d [② / ① , λ / λ] = {▨ / ▨}, φ_c^d [① / ▨ , λ / λ] = {▨ / ▨}.

A sample computation to generate the following picture

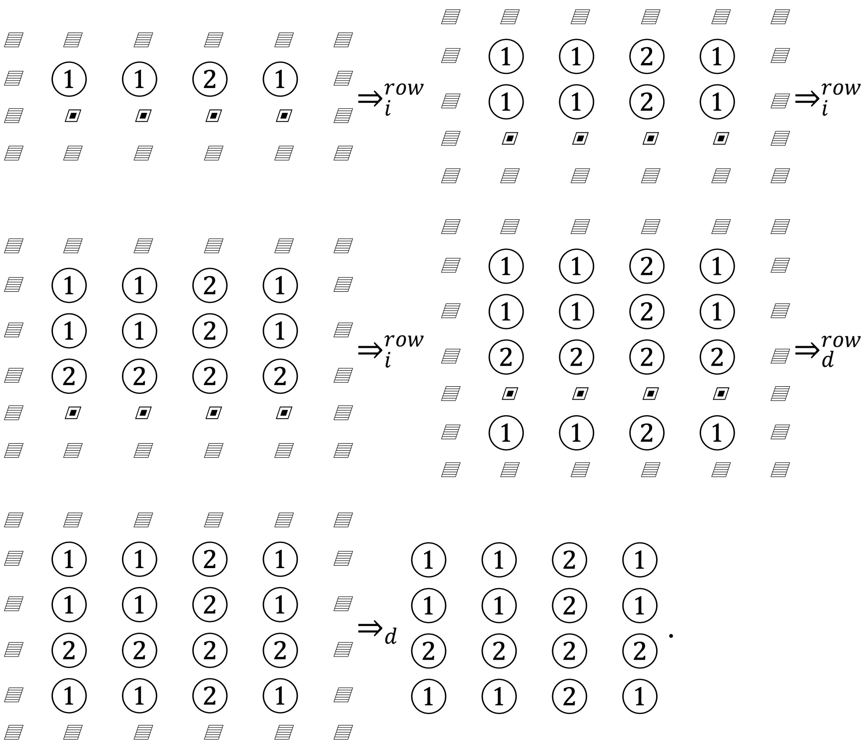
①	①	②	①
①	①	②	①
②	②	②	②
①	①	②	①

is given below:

$M \rightarrow$ ▨ ▨ ▨ ▨ / ▨ ① ▫ ▨ $\Rightarrow_i^{col}$ ▨ ▨ ▨ ▨ ▨ ▨ / ▨ ① ① ② ▫ ▨ $\Rightarrow_i^{col}$

▨ ▨ ▨ ▨ ▨ ▨ ▨ / ▨ ① ① ② ▫ ① ▨ $\Rightarrow_d^{col}$ ▨ ▨ ▨ ▨ ▨ ▨ / ▨ ① ① ② ① ▨ $\Rightarrow_i^{row}$

Published by: The Mattingley Publishing Co., Inc.



Working description is as follows:

Beginning with axiom set picture, column insertion rules are applied so as to expand the picture horizontally by inserting appropriate columns at the appropriate positions. After horizontal expansion, the nonterminal column involving ▢'s is to be deleted using suitable deletion rules and then row insertion rules are applied suitably at apposite position so as to expand the picture vertically and then the rows involving the non-terminal ▣ is to be removed using row deletion rules and finally the border symbol ▤ is to be deleted using both row and column deletion rules to get the required picture.

CONCLUSION

In this paper we have shown the non-empty intersection of L(PCAIDG) with L(TRG) and L(RTG). There is scope for further research in the direction of proving L(PCAIDG) properly includes L(TRG) since languages generated by tile rewriting grammars are of context free type. Moreover, we try to establish a hierarchy, like the Chomsky hierarchy for string grammars, for two-dimensional grammars generating picture language.

REFERENCES

[1] D. Giammarresi, and A. Restivo: Two-dimensional Languages, Handbook of formal languages, vol.3, 215-267 (1997).

[2] D.G. Thomas, S. James Immanuel, Atulya K. Nagar and T. Robinson: Parallel Contextual Array Insertion Deletion Grammar, Lecture Notes in Computer Science 11255, 28-42 (2018).

[3] S. CrespiReghizzi, M. Pradella: Tile Rewriting Grammars and Picture Languages: Theoretical Computer Science, 340(2), 257-272 (2005).