Bulletin of Pure and Applied Sciences
Section - E - Mathematics & Statistics
Website : https : //www.bpasjournals.com/

Bull. Pure Appl. Sci. Sect. E Math. Stat.
38E(Special Issue)(2S), 12–17 (2019)
e-ISSN:2320-3226, Print ISSN:0970-6577
DOI: 10.5958/2320-3226.2019.00072.9

Generation of $C_m \cdot P_n^l$ graphs using P system *

Emerald Princess Sheela J.D.[1], Thanga Murugeshwari V.[2], J. Jeba Jesintha[3] and C.D. Devadharshini[4]

1, 2. Department of Mathematics, Queen Mary's College,
Affiliated to University of Madras, Chennai, India.
3, 4. P.G. Department of Mathematics, Women's Christian College,
Affiliated to University of Madras, Chennai, India.
1. E-mail: emeraldsolomon96@gmail.com , 3. E-mail: jjesintha_75 @yahoo.com

Abstract Graph grammars, used for parsing and generating graphs, provide a useful formalism for describing structural manipulation of multi-dimensional data. Hence graph grammars are used as a synonym for graph rewriting system, especially in the context of languages. Hyperedge replacement graph grammars, edge replacement graph grammars and node replacement graph grammars are the three types of graph replacement techniques in which hyperedge replacement graph grammar is used in this paper. P system is a membrane structure consisting of objects and evolution rules in it. The objects can be transformed from on membrane to the other or can dissolve the membrane. In this paper we generate $C_m \cdot P_n^l$ graphs for $m \geq 3, n \geq 2, l \geq 2$ using the P system.

Key words Graph Grammars, P system, hyperedge replacement, conditional communication.

2010 Mathematics Subject Classification Primary: 05C38, 05C65, 05C90, 05C99.

1 Introduction

Graph grammars are used in manipulating multidimensional patterns which cannot be done easily in string grammars. It was introduced by D. Janssens and G. Rozenberg. Graph grammar consists of a mother graph, a daughter graph and an embedding mechanism. By applying the embedding mechanism graphs can be generated. Graph grammar has wide applications in pattern recognition, e.g. see, Engelfriet [2].

Hyperedge replacement graph grammars are used to derive hypergraphs from the production rules which generates hypergraph language theory. It has good structural and algorithmic properties. The graph $C_m \cdot P_n^l$ is the product of cycle C_m with the path P_n^l of length n with multiplicity l. This graph has applications in network theory especially in neural networks [3].

The P system was the concept introduced in 1998 by Gheorghe Păun [8], whose last name is the origin of the letter "P" in the P system. The P system is concerned with the use of a computational model.

* Communicated, edited and typeset in Latex by *Lalit Mohan Upadhyaya* (Editor-in-Chief).
Received March 16, 2019 / Revised November 14, 2019 / Accepted December 12, 2019. Online First Published on June 03, 2020 at https://www.bpasjournals.com/.
Corresponding author J. Jeba Jesintha, E-mail: jjesintha_75@yahoo.com
Refereed Proceedings of the National Conference on Mathematics and Computer Applications held at the Department of Mathematics, Women's Christian College, Chennai, India from January 29, 2019 to January 30, 2019.

Membranes are the main "structure" in a P system, it contains a set of rules, symbols and conditions within itself (see, Păun [4]). The outer most membrane is considered as a container membrane or skin membrane. The system is widely used in the field of computer science for manipulation purpose. It has applications in DNA study, epidemiology and taxonomy (see, [1,5]).
In this paper, we generate the Actena graph and the $C_m \cdot P_n^l$ graph using hyperedge replacement graph P system with conditional communication.

2 Preliminaries

Definition 2.1. [7] Let K be an arbitrary but finite set of labels and let $'\text{type}'$ be a typing function from K to the set of natural numbers $\mathbb{N}$. A hypergraph H over K is a tuple

$$(\text{Vertices, Hyperedges, attach, labelling, external})$$

where attach : Hyperedges $\rightarrow$ Vertices* is a mapping that assigns a sequence of pair wise distinct attachment node attach (e) to each hyperedge e, labelling is a mapping from hyperedges to K that labels each hyperedge such that the type $(\text{labelling}\,(e)) = |\text{attach}\,(e)|$, external $\in$ Vertices* is a sequence of pair wise distinct external nodes. H_k denotes set of all hypergraphs over K.

Definition 2.2. [7] A ***hyperedge replacement grammar*** is a system $\text{HRG} = (N, T, P, S)$, where $N \subseteq K$ is a set of non-terminals. $T \subseteq K$ with $T \cap N = \emptyset$ is a set of terminals; P is a finite set of productions. A production over N is an ordered pair $P = (A, R)$ with $A \in N, R \in H_k$ and $\text{type}\,(A) = \text{type}\,(R)$. A is called the left-hand side of P and is denoted by $\text{lhs}\,(P)\text{and}R$ is called the right-hand side of Pwhich is denoted $\text{rhs}\,(P)$. $S \in N$ is a start symbol. The class of all hyperedge replacement grammars is denoted by HRG.

Definition 2.3. [7] A ***$m-$ hypergraph*** is defined as a hypergraph with m external nodes and a handle e (single hyperedge) with attach $H\,(e) =$ external H. If labelling $H\,(e) = A$, then H is said to be handle induced by A and is denoted by A^*.

Definition 2.4. [7] The ***hypergraph language*** $\text{L}\,(\text{HRG})$ generated by HRG is $\text{L}_s\,(\text{HRG})$ where $\text{L}_s\,(\text{HRG})$ consists of all hypergraphs derivable from the start symbol S.

Definition 2.5. [6] A ***hyperedge replacement graph P system with conditional communication*** (HRGPCC) is a construct

$$\Pi = (V, T, \mu, M_1, M_2, \ldots, M_n, (R_1, P_1, F_1), (R_2, P_2, F_2), \ldots, (R_n, P_n, F_n), (n, d))$$

where, V is a finite set of non-terminal hyperedge labels and terminal symbols; T is a set of terminal symbols available in V; μ is a membrane structure with n membranes and depth d, which are injectively labeled by numbers in the set $\{1, 2, \ldots, n\}$. The skin membrane is labeled as 1, M_i is the finite set of hyperedges over V initially present in the region $i, i = 1, 2, \ldots, n$ of the system, R_i is the finite of hyperedge rules associated with the region $i, i = 1, 2, \ldots, n$. The parallelism mode works as follows:
P_i and F_i are permitting and forbidding condition associated with the region $i, i = 1, 2, \ldots, n$ of the form empty or symbol checking given as follows:

1. Empty: no restriction is imposed on graphs, they either exit the current membrane or enter any of the directly inner membranes freely: we denote an empty permitting condition by $(\text{true, in}), (\text{true, out})$ and an empty forbidding condition by $(\text{false, notin}), (\text{false, notout})$.
2. Symbols checking: each P_i is a set of pairs $(A, \alpha), \alpha \in \{\text{in , out}\}$ for $A \in V$ and each F_i is a set of pairs $(B, \text{not } a), a \in \{\text{in , out}\}$ for $B \in V$; a graph W can go to a lower membrane only if there is a pair $(A, \text{ in}) \in P_i$ with $A \in \text{labelling}\,(W)$ and for each $(B, \text{ notin}) \in F_i, B \notin \text{labelling}\,(W)$. Similarly, the graph goes out of a membrane i, if there is at least one pair $(A, \text{ out}) \in P_i$ and $(B, \text{ notout}) \in F_i$ for all $B \notin \text{labelling}\,(W)$.

The family of languages generated by hyperedge replacement graph P systems with conditional communication is denoted by $\text{HRGPCC}_n^d\,(\alpha, \beta)$, $\alpha, \beta \in \{\text{empty , symbol}\}$ denote the permitting and forbidden rules.

Definition 2.6. [5] A ***rewriting step with unique parallelism*** (U) involves the substitution of all occurrences of exactly one symbol according to exactly one rule, which is non-deterministically chosen between all rules that can be applied to that symbol. That is, given a string $w = x_1ax_2a\ldots x_nax_{n+1}$ with $x_i \in (V\backslash\{a\})*, \forall i = 1, 2, \ldots, n+1$ and one context-free rule $r : a \to \alpha$ we obtain the string $w' = x_1\alpha x_2\alpha\ldots x_n\alpha x_{n+1}$ in one parallel rewriting step.

In this paper we generate a language that consists of cycle C_m in which a caterpillar is appended with each m^{th} vertex. We call the above language L(HRG).

3 Generation of L(HRG) using hyperedge replacement graph grammar

Let us consider a hyperedge replacement grammar $\text{HRG} = \{\{D, E, K\}, \{\text{Prd}_1, \ldots, \text{Prd}_6\}, S\}$ where we have the following six various productions in the form of graphs as depicted in Fig. 1.

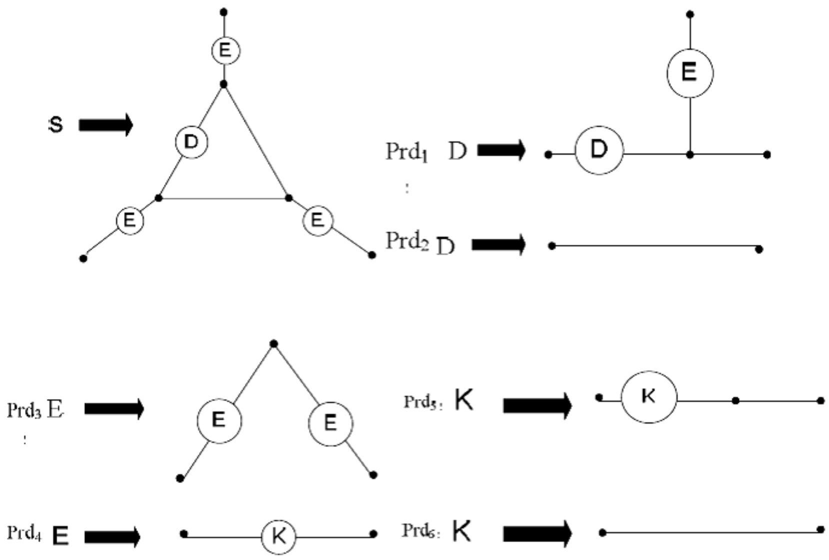

Fig. 1: Generating graphs using HRG.

In this replacement S will be a start point. This S can be non-deterministically replaced by any of the six $\text{Prd}_1$to Prd_6 from which we obtain various types of cycles C_m with m rooted trees in HRG.
In Fig. 2 below first the axiom S is applied, then Prd_1 is applied in which the non-terminal D is replaced which adds a non-terminal E to the existing graphs. The application of Prd_2 vanishes D with a path of length 2. In the third step Prd_3 is applied to one of the non-terminal E and the same is repeated for another non-terminal E. In the resultant graph Prd_4 is applied to each E six times to get a graph which consists of the non-terminal K. Now P_5 is applied four times to K and then P_6 is applied to all the K's that vanishes all the non-terminals from the graph.
The language generated by the above HRG is $\text{L}(\text{HRG}) = \text{Cycle } C_m$ in which each m^{th} vertex is a caterpillar.

4 Generation of $C_m \cdot P_n^l$ graph using hyperedge replacement graph Psystem with conditional communication

In this P system we consider two membranes, the outer membrane consists of the start symbol, 2 conditions, 3 forbidden rules and 2 permitting rule and the inner membrane consists of 1 condition, 1 forbidden rule and 1 permitting rule.
Example 4.1. Consider a P system for which (see Fig. 3)

$$\Pi = \left\{S, D, E, K, P_2, [_1[_2]_2]_1, S, (R_1, P_1, F_1)(R_2, P_2, F_2)(2, 2)\right\}.$$

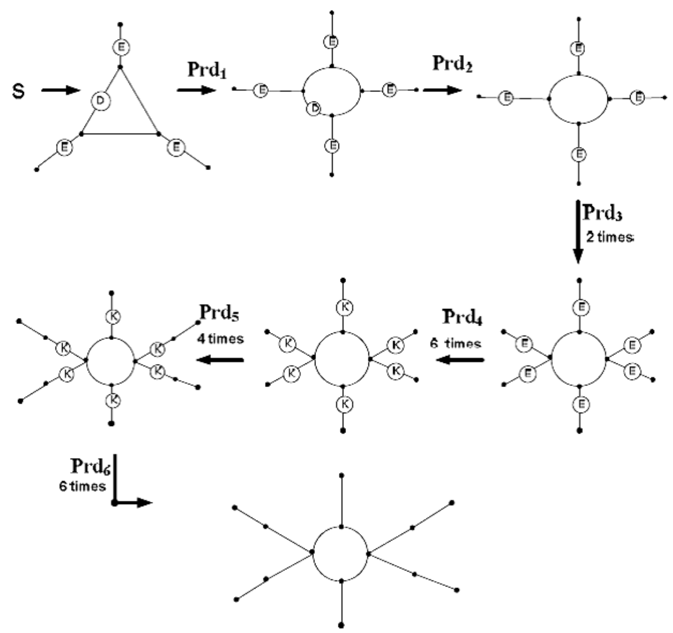

Fig. 2: Generation of cycle C_4 with 4 rooted trees using *HRG*.

4.1 Generation of $C_4 \cdot P_3^2$ graphs in HRG in Psystem

Initially the start axiom S is applied then D is replaced by Prd_1 which introduces a non-terminal E to the existing graph. When $\mathrm{Prd}_2$is applied to this structure D is replaced by a path of length 2. Now the forbidding and permitting conditions are satisfied for D which moves the graph to the membrane 2. After that Prd_3 is applied to E, since unique parallelism mode is used all E's are simultaneously replaced. The application of Prd_4 changes all E's to K. Again forbidding and permitting conditions are satisfied which in turn moves the graph to the outer membrane. Here the rules Prd_5 and Prd_6 are applied. The graph obtained by using the above productions generates $C_4 \cdot P_3^2$ (Fig. 4). Instead of applying Prd_5 if we directly replace the graph by Prd_6we obtain the Actena graph as shown in Fig. 5.

When we apply the rule 1 five times and the rule 2 one time we get a cycle of length 8, which on further applying rule 3 one time, rule 4 one time produces a cycle of length 4 to which a further application of the rule 5 six times and the rule 6 one time extends the graph to path 7, then the resultant graph will be $C_8 \cdot P_7^4$ graph which is shown in Fig. 6.

The hyperedge replacement graph P system with conditional communication with 1 membrane is contained in the hyperedge replacement graph P system with conditional communication with 2 membranes.

Consider the P system in Example 4.1, the graph $C_m \cdot P_n^l$ cannot be generated in lesser membrane, since, if the growth of D is stopped by applying the path Prd_2 in production 3 it enters the second membrane in which E grows evenly (unique parallelism). This movement cannot be done in single membrane where the length of E will be varied.

5 Conclusion

In this paper we are able to generate the Actena graph and the $C_m \cdot P_n^l$ graph by using the hyperedge replacement graph rewriting P system. We focus on the generation of graphs which are of practical importance, by using different graph grammars and their corresponding P systems.

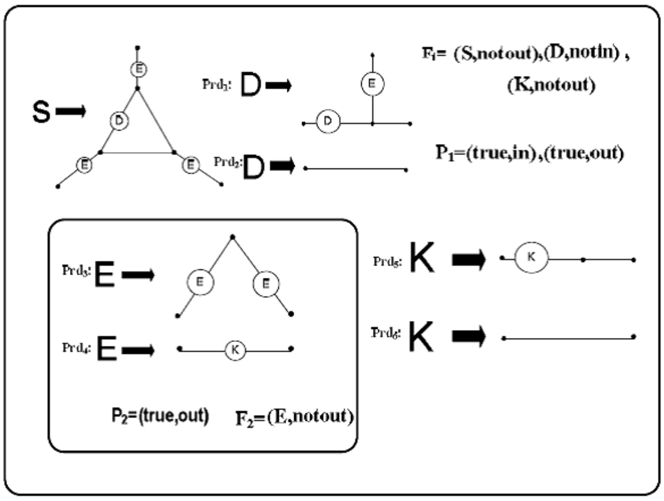

Fig. 3: Membrane structure in P system.

References

[1] Bottoni, P., Labella, A., Martin-Vide, C. and Paun, G. (2002). Rewriting P system with conditional communication, *Lecture Notes in Computer Science*, 2300, 325–353.

[2] Engelfriet, J. (1997). Context free graph grammars, in: Rozenberg, G. and Salomaa, A. (Eds.) Handbook of Formal Languages, Springer, Berlin, Heidelberg.

[3] Habel, A. and Kreowski, H. (1987). May we introduce to you: Hyperedge Replacement, *Lecture Notes in Computer Science*, 291, 15–26.

[4] Păun, G. (2002). A guide to membrane computing, *Theoretical Computer Science*, 287, 73–100.

[5] Paun, G., Rozenberg, G. and Saloma, A. (2010). The Oxford Handbook of Membrane Computing, Oxford University Press, Oxford.

[6] Sankar, Meena Parvathy, David, N.G. and Thomas, D.G. (2011). Hyperedge replacement graph P system, Sixth International Conference on Bio-Inspired Computing Theories and Applications, 27–29 Sept. 2011, Penang, Malaysia.

[7] Rozenberg, G. (1997). Handbook of Graph Grammars and Computing by Graph Transformation, World Scientific, New York, Singapore.

[8] Păun, Gheorghe G. (2000). Computing with membranes, *Journal of Computer and System Sciences*, 61(1), 108–143.

[9] Janssens, D. and Rozenberg, G. (1983) Graph grammars with node-label controlled rewriting and embedding, in Ehrig, H., Nagl, M. and Rozenberg, G. (Eds.) Graph-Grammars and Their Application to Computer Science. Graph Grammars 1982, Lecture Notes in Computer Science, Vol. 153, Springer, Berlin, Heidelberg.

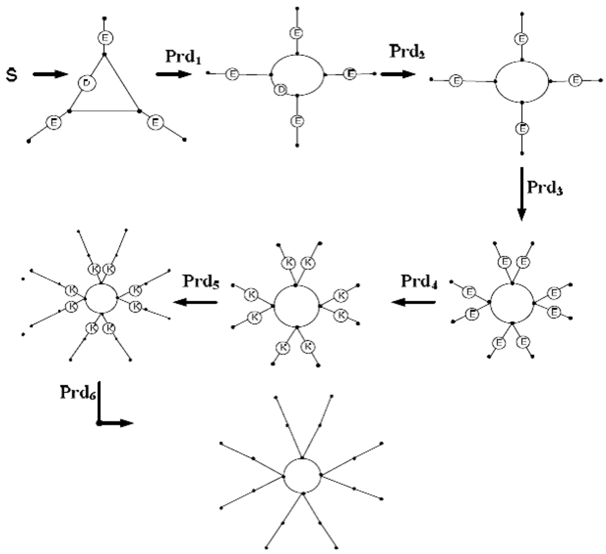

Fig. 4: Generation of $C_4 \cdot P_3^2$ graph using HRG in P system.

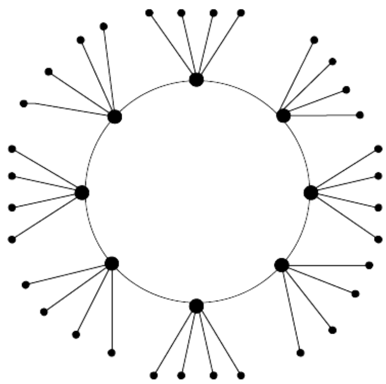

Fig. 5: Actena Graph.

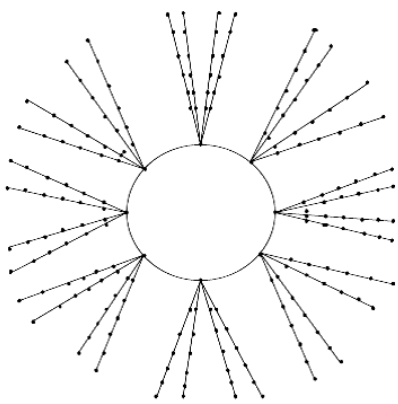

Fig. 6: $C_8 \cdot P_7^4$ graph.