International Journal of Applied Graph Theory
Vol.1, No.2 (2017), 53 - 73.
$ISSN(Online) : 2456 - 7884$

Generating and Coloring Some Family of Graphs Using Hypergraph Grammar

S. Therasa
Department of Mathematics
University College of Engineering
BIT Campus, Anna University, Tiruchirappalli
Tamilnadu, India.
thema.jeje@gmail.com

T. Rajaretnam
Department of Mathematics
St.Joseph's College (Autonomous), Tiruchirappalli
Tamilnadu, India.
t_rajaretnam@yahoo.com

Abstract

This study deals with the concept of hyper graph grammar which results in regular multi-hypergraph language. It allows the terminal arcs of the generated graph to multiply in number though the skeleton may be crucial. We have used hypergraph grammar as a tool to generate some family of graphs that are familiar in the field of graph theory. We have also written syntax directed translation scheme in the production for coloring the generated graphs as an application of hypergraph grammar.

Keywords: Hypergraph grammar, regular multi-hypergraph, regular multi-hypergraph language.

2010 Mathematics Subject Classification : 03D05, 05C85

1 Introduction

Graph grammar originated in the late 60's, motivated as a promising tool for picture processing problems. Graph grammars have been developed as an extension of the concept of formal grammars on strings to grammars on graphs[3]. D.Caucal focused on providing some of the basic tools to reason out deterministic graph grammar and on structural

* Corresponding Author: $S.Therasa$
Ψ Received on June 22, 2017 /Revised on October 21, 2017/ Accepted on October 21, 2017

study of their generated graphs[1,2]. In graph theory, graph coloring is a special case of graph labeling. It is an assignment of labels traditionally called "colors" to elements of a graph subject to certain constraints. Graph coloring has been studied as an algorithmic problem since the early 1970s. One of the major applications of graph coloring, register allocation in compilers, was introduced in 1981[4].

This study deals with the concept of hypergraph grammar which is similar to string generating grammar whereas it generates multi-hypergraph instead of words. It defines multi-hypergraph which is a collection of multi-hyperarcs. The number of occurrences of any arc is multiple in number and is also finite. It is called as multiplicity of the arc. It further extends that hypergraph grammar $R = (G_0, P)$ is an ordered pair where G_0 is an initial graph and P is a set of collection of rules of the form $X \to H$ or $H \to H'$. Here X is a hyperarc, H and H' are the multi- hypergraphs respectively. It is also concerned with the specific setting where the considered rules are deterministic and context free. Deterministic means that there is only one rule for every non-terminal. It further extends its studies in the generated graph that, the number of occurrences of a terminal hyperarc be multiple in numbers even though the skeleton would be more crucial. It also discusses about regular multi-hypergraph of a given grammar and the languages generated by the grammar too. The classification over the language is made with respect to orientation and the multiplicity of the arc.

In this paper, Hypergraph grammar is used to generate some special family of graphs that are familiar in graph theory. We have also used syntax directed translators for coloring the graphs which adapts hypergraph grammar as a tool.

2 Basic Definitions

In this section, we have reviewed some fundamental concepts related to hypergraph grammar.

A finite set E of symbols is an alphabet of letters. E^* is the set of words over E. Any word $u \in E^n$ is of length $|u| = n$ is also represented by a mapping from $\{1, 2, 3, \ldots, n\}$ into E or by the juxtaposition of its letters $u = u(1)u(2)u(3)\cdots u(n)$.

Definition 2.1. A multi-subset M of E is a mapping from E into $\mathbb{N}$ where for any $e \in E$, the integer $M(e)$ is its multiplicity, the number of occurrences of e in M. It is also represented by the functional subset $\{\Big(e, M(e)\Big) \mid e \in E \wedge M(e) \neq 0\}$ of $E \times \mathbb{N}_+$.
If (e, m), $(e, n) \in M$ then $m = n$. The cardinality of M is $|M| = \sum\limits_{e \in E} M(e)$.
M is said to be finite if its support $\hat{M} = \{e \in E \mid M(e) \neq 0\}$ is finite.

Definition 2.2. Let F be a set of symbols called labels ranked by a mapping $\varrho : F \to \mathbb{N}$ associating to each label f its arity $\varrho(f)$ and such that $F_n = \{f \in F \mid \varrho(f) = n\}$ is countable $\forall n \geq 0$.

Definition 2.3. A simple, oriented and labeled hypergraph G is a subset of $\bigcup_{n\geq 0} F_n V^n$ where V is an arbitrary set such that its vertex set

$$V_G = \{v \in V \mid FV^*vV^* \cap G \neq \emptyset\}$$

is finite or countable. Its label set $F_G = \{f \in F \mid FV^* \cap G \neq \emptyset\}$ is finite.
Any $fv_1v_2 \ldots v_{\varrho(f)}$ is a hyperarc labeled by f and of successive vertices $v_1, v_2, \ldots, v_{\varrho(f)}$. If $\varrho(f) \geq 2$ then it depicts an arrow labeled f and successively linking $v_1, v_2, \ldots, v_{\varrho(f)}$. If $\varrho(f) = 1$ then it depicts a label of f on vertex v_1 and f is called a color of v_1. If $\varrho(f) = 0$ then it depicts an isolated label f called a constant.

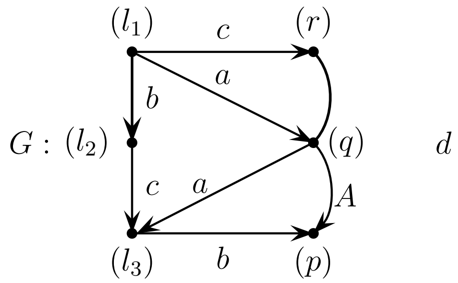


Figure 2.3

$G = \{bl_1l_2,\ cl_2l_3,\ cl_1r,\ al_1q,\ aql_3,\ bl_3p,\ Arqp,\ d\}$; $V_G = \{l_1, l_2, l_3, p, q, r\}$; $F_G = \{a, b, c, A, d\}$; $F_3 = \{A\}$; $F_2 = \{a, b, c\}$; $F_0 = \{d\}$.

Definition 2.4. The transformation of a hypergraph G by a function h from V_G into a set V is the hypergraph $h(G) = \{fh(v_1)h(v_2) \cdots h(v_{\varrho(f)}) \mid fv_1v_2 \cdots v_{\varrho(f)} \in G\}$.
For the graph given in figure 2.3, h(G) is shown below if the transformation h is defined as $h(l_1) = m_1; h(l_2) = m_2; h(l_3) = m_3; h(p) = p_1; h(q) = q_1; h(r) = r_1$.

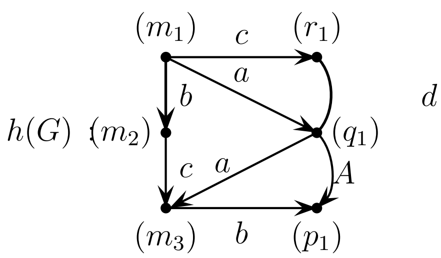


Figure 2.4

Definition 2.5. An isomorphism h from a hypergraph G to a hypergraph H is a bijection from V_G to V_H such that $h(G) = H$ and write $G \overset{h}{\sim} H$ or $G \sim H$ if we do not specify the bijection.

Definition 2.6. A multi-hypergraph G is a multi-subset of $\bigcup_{n\geq 0} F_n V^n$ where V is an arbitrary set. Each arc $X \in G$ depicted $G(X)$ times. The vertex set V_G and the label set F_G of a multi-hypergraph G are the sets defined on its support $\hat{G}$. $V_G = V_{\hat{G}}$ and $F_G = F_{\hat{G}}$.

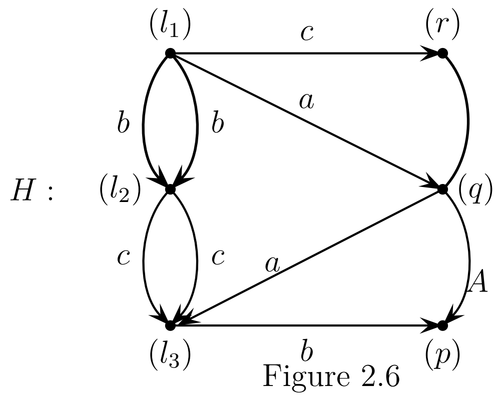


Figure 2.6

Definition 2.7. The transformation of any multi-hypergraph G by any function h from V_G into a set is extended as $\big(h(G)\big)(X) = \sum_{h(Y)=X} G(Y)$ for any hyperarc X, assuming that the sum is always finite.

Definition 2.8. A hypergraph grammar is a finite set of rules of the form $fx_1x_2x_3\cdots x_{\varrho(f)} \to H$ where $fx_1x_2x_3\cdots x_{\varrho(f)}$ is a hyperarc joining pairwise distinct vertices $x_1 \neq x_2 \neq x_3 \neq \cdots x_{\varrho(f)}$ and H is a finite multi- hypergraph. The labels of the left hand sides of the rules of the grammar are the non-terminals of R and denoted by $N_R = \{f \in F \mid fX \in Dom(R)\}$. The labels of R which are not non-terminals are the terminals of R and denoted by $T_R = \{f \in F - N_R \mid \exists P \in Im(R), fX \in P\}$. $F_R = N_R \cup T_R$ be the labels of R and $\varrho(R) = Max\{\varrho(A) \mid A \in N_R\}$ be the arity of R.

Definition 2.9. A deterministic hyper graph grammar means that there is only one rule for every non-terminal.(i.e.,) if $(X, H), (Y, K) \in R$ and $X(1) = Y(1)$ then $(X, H) = (Y, K)$

Definition 2.10. For any rule (X, H), we say that $V_X \cap V_H$ are the inputs of H and $\cup\{V_Y \mid Y \in H \wedge Y(1) \in N_R\}$ are the outputs of H.

Notation 2.11. We write a hyperarc as the word fY, where f is its label and Y its vertex word. It can also be represented as X where the first letter $X(1)$ is its label and for $1 \leq i \leq |X|$, the ith letter $X(i)$ is its $(i-1)$th vertex. We use upper case letters $A, B \cdots$ for non terminals and lower case letters $a, b, c \cdots$ for terminals. A hypergraph grammar R is said to be graph grammar if the terminals are of arity 1 or 2. For any rule R_1 of the grammar R, $Dom(R_1)$ and $Im(R_1)$ depicts the left and the right hand side of the rule R_1 respectively.

Definition 2.12. Let $G = (V, E)$ be a graph and let $C = \{c_1, ..., c_k\}$ be a finite set of colors (labels). A vertex coloring is a mapping $c : V \to C$ with the property that if $(v_1, v_2) \in E$, then $c(v_1) \neq c(v_2)$.

Definition 2.13. A graph $\mathrm{G} = (\mathrm{V}, \mathrm{E})$ is k-colorable if there is a vertex coloring with k colors.

Definition 2.14. Let $G = (V, E)$ be a graph. The chromatic number of G written $\chi(G)$ is the minimum integer k such that G is k-colorable.

2.1 Derivation

In this section, we define hypergraph grammar suitably and propose two methods in the derivation of the hypergraph grammar R.

Definition 2.15. A hypergraph grammar $R = (G_0, P)$ is an ordered pair where G_0 is the base graph and P is a finite set of rules. Each rule of P is of the form $fx_1x_2x_3 \cdots x_{\varrho(f)} \to H$ or $H \to H'$ where $fx_1x_2x_3 \cdots x_{\varrho(f)}$ is a hyperarc joining pairwise distinct vertices $x_1 \neq x_2 \neq x_3 \neq \cdots x_{\varrho(f)}$ and H, H' are finite multi-hypergraphs. The labels of the left hand side of the rules of the grammar are the non-terminals of R and denoted by $N_R = \{f \in F \mid fX \in Dom(R)\}$. The labels of R which are not non-terminals are the terminals of R and denoted by $T_R = \{f \in F - N_R \mid \exists P \in Im(R), fX \in P\}$. $F_R = N_R \cup T_R$ be the labels of R and $\varrho(R) = Max\{\varrho(A) \mid A \in N_R\}$ be the arity of R.

Method 1: A multi-hypergraph M derives N written $M \underset{R,X}{\longrightarrow} N$ if we choose a non-terminal hyperarc X in M where $X = As_1s_2 \ldots s_{\varrho(A)}$ and a rule $X' \to H$ in R where $X' = Ax_1x_2 \ldots x_{\varrho(A)}$ in R such that N can be obtained by replacing X by H in M. Thus, $N = (M - X) + h(H)$ for some function h, mapping each x_i to s_i and other vertices of H injectively to vertices outside of M. For any hyperarc Y,

$$N(Y) = M(Y) + \Big(h(H)\Big)(Y) \text{ if } Y \neq X$$

$$N(X) \;=\; \Big(M(X)-1\Big)+\Big(h(H)\Big)(X)$$

Method 2: The derivation $\underset{R,X}{\longrightarrow}$ of a hyperarc X is extended in an obvious way to the derivation $\underset{R,E}{\longrightarrow}$ of any multi-subset E of non-terminal hyperarcs. The complete derivation $\Rightarrow$ is the derivation according to the multi-subset of all non-terminal hyperarcs. $M[R]N$ if $M \underset{R,E}{\longrightarrow} N$. where E is the multi-subset of all non-terminal hyperarcs of M.

Here, $M \underset{R,E}{\longrightarrow} N$ means that N is obtained from M by replacing each non-terminal hyperarc in E sequentially by its corresponding multi-hypergraph. Suppose that,
$E=\Big\{\Big(X_1,M(X_1)\Big),\Big(X_2,M(X_2)\Big),\Big(X_3,M(X_3)\Big),\cdots\Big(X_n,M(X_n)\Big)\Big\}$ be a multi-subset of all non-terminal hyperarcs of M. Then, the k^{th} occurrence of X_i be replaced by $g_{\{i,j,k\}}(H_j)$ where H_j being a multi-hypergraph of a rule $Y_j \to H_j$ of R and $g_{\{i,j,k\}}$ being a transformation such that $g_{\{i,j,k\}}(Y_j)=X_i, \forall k=1,2,3,\ldots M(X_i)$.
Thus, $N=(M-E)+\sum\limits_{(i,j,k)\in M_1} g_{\{i,j,k\}}(H_j)$ where $M_1=(i,j,k) \mid \exists\, Y_j \to H_j \in R,\, X_i \in M,\, X_i(1)\in N_R$ and $g_{\{i,j,k\}}(Y_j)=X_i$

If the n consecutive ($n \neq 1$) sequence of parallel rewriting yields multi-hypergraphs $G_1,G_2,G_3\cdots G_n$ from an initial multi-hypergraph G_0 using a hypergraph grammar R then it would be written $G_0[R]G_1[R]G_2[R]G_3\ldots[R]G_n$. Here G_n is said to be derivable from $G_0[R]^*G_n$.

Notation 2.16. For a given hypergraph M, $[M]=M\cap T_R V_M^*$ designates the simple set of terminal hyperarcs of M.

Notation 2.17. For a given multi-hypergraph M, $\sqsubset M \sqsupset = M\cap T_R V_M^*$ designates the multi-set of terminal hyperarcs of M

3 Regular multi-hypergraph

In this section, we define regular multi- hypergraph, undirected regular multi- hypergraph and the hypergraph languages generated by the above grammar.

Definition 3.1. A multi-hypergraph G is generated by a hypergraph grammar R from a multi-hypergraph H if G is isomorphic to a hypergraph in the following set called multi hypergraph language.

$$R(H)=\left\{\bigcup_{n\geq 0} \sqsubset H_n \sqsupset \mid H_0=H \wedge \forall n\geq 0\ H_n \Rightarrow H_{n+1}\right\}$$

Definition 3.2. A regular multi- hypergraph is a multi-hypergraph generated by a deterministic hypergraph grammar from a finite multi-hypergraph.

For any multi-hypergraph H, $\ll H \gg$ designates the multi-set of undirected terminal hyperarcs. For any hypergraph H, $\parallel H \parallel$ designates the simple set of undirected terminal hyperarcs.

Definition 3.3. An undirected multi-hypergraph G is generated by a hypergraph grammar R from a multi-hypergraph H if G is isomorphic to a hypergraph in the following set called undirected multi-hypergraph language.

$$\overline{R}(H) = \left\{ \bigcup_{n\geq 0} \ll H_n \gg \mid H_0 = H \wedge \forall n \geq 0 \ H_n \Rightarrow H_{n+1} \right\}$$

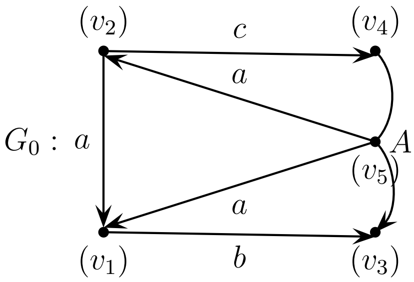
Definition 3.4. A hypergraph G is generated by a hypergraph grammar R from a multi-hypergraph H if G is isomorphic to a hypergraph in the following set called hypergraph language.

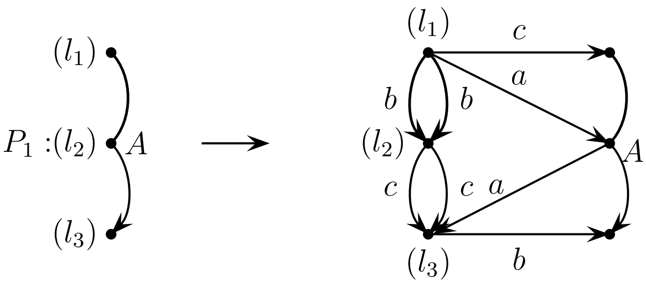
$$R^*(H) = \left\{ \bigcup_{n\geq 0} [H_n] \mid H_0 = H \wedge \forall n \geq 0 \ H_n \Rightarrow H_{n+1} \right\}$$

Definition 3.5. An undirected hypergraph G is generated by a hypergraph grammar R from a multi-hypergraph H if G is isomorphic to a hypergraph in the following set called undirected hypergraph language.

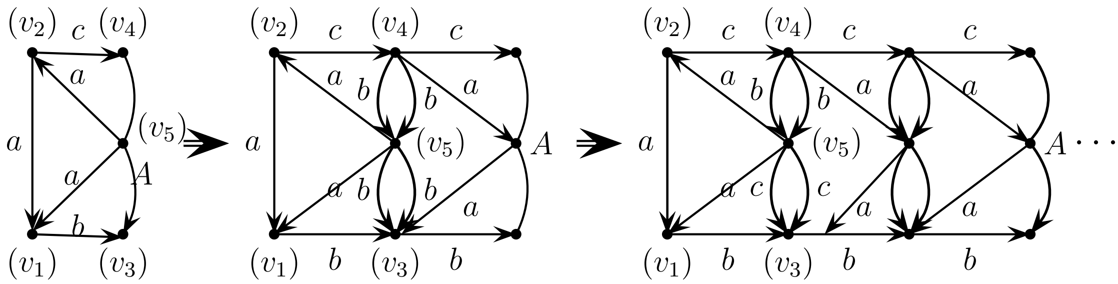
$$R^\star(H) = \left\{ \bigcup_{n\geq 0} \parallel H_n \parallel \mid H_0 = H \wedge \forall n \geq 0 \ H_n \Rightarrow H_{n+1} \right\}$$

Example 3.6. Let us consider an initial graph G_0 and the rule P_1 as given in the following figure. In this example, we have shown the steps of derivation and the possible languages that are generated by the grammar.

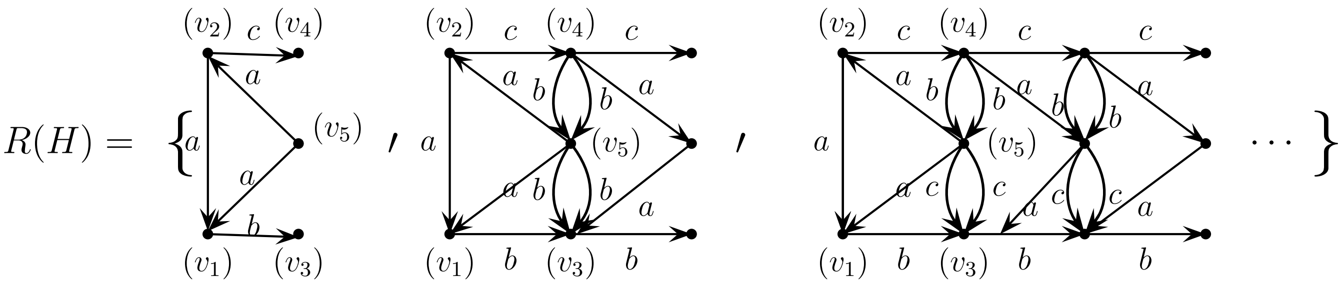


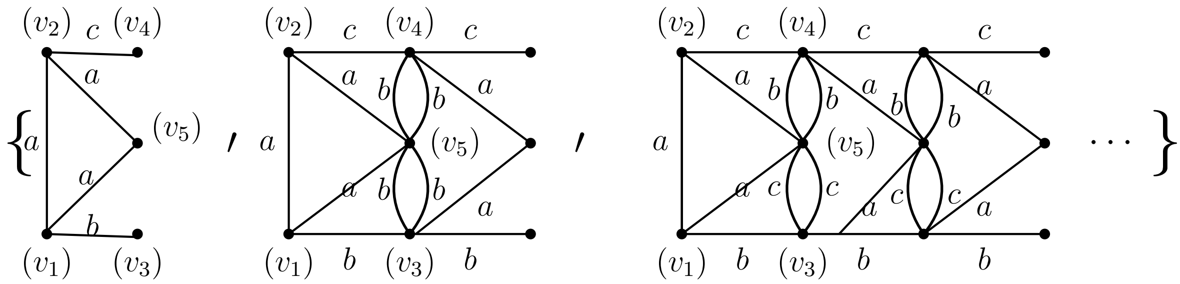


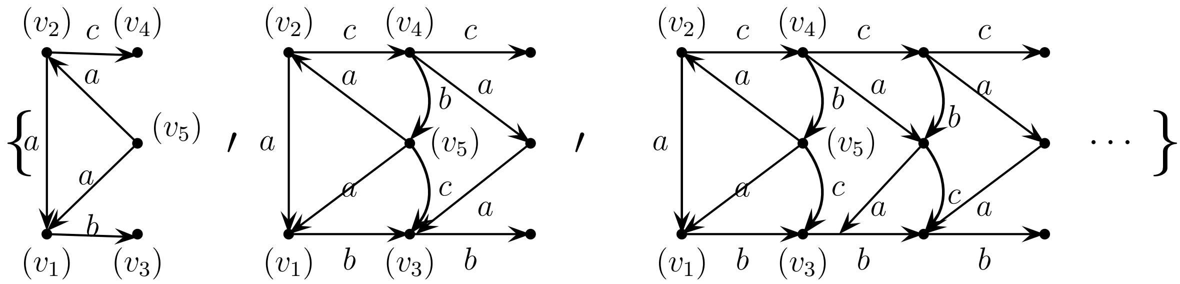
Derivation :

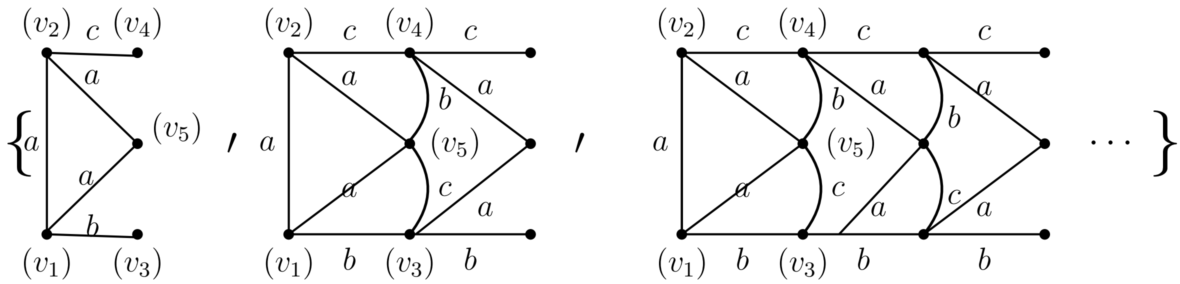


The languages generated by the above grammar are:



$\overline{R}(H) =$ 

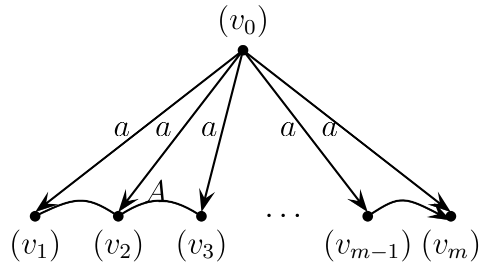
$R^{*}(H) =$ 

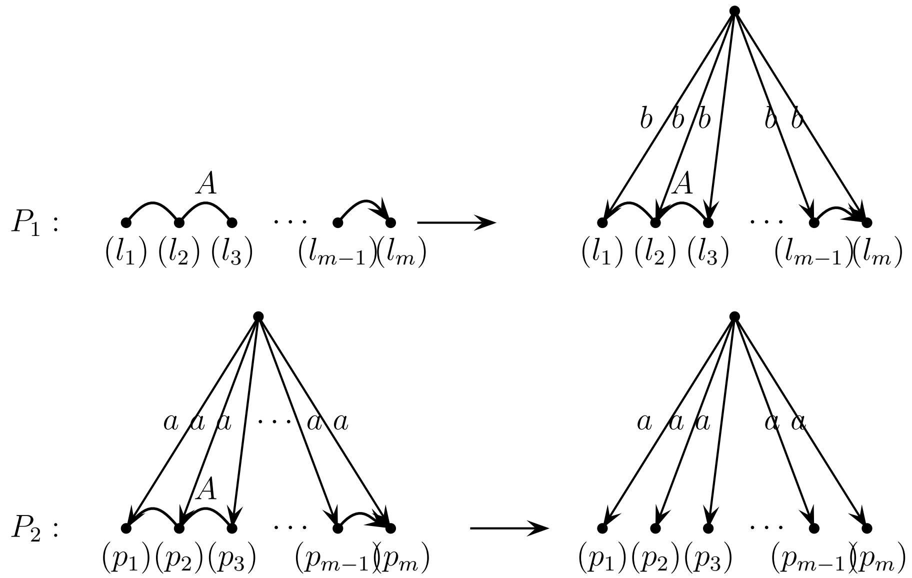
$R^{\bigstar}(H) =$ 

4 Generation of Graphs

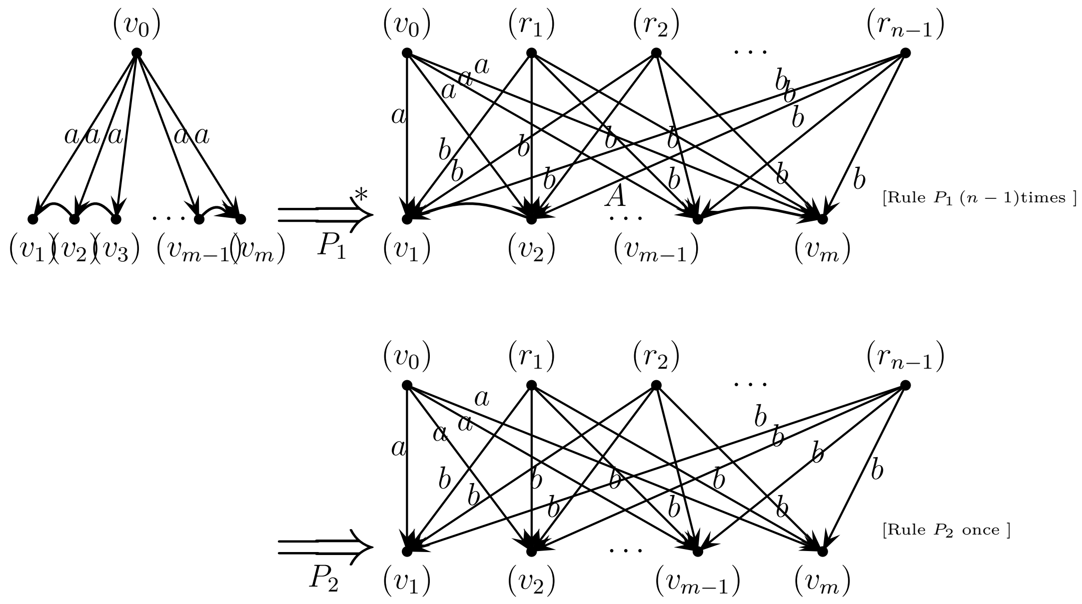
In this section, we construct grammars to generate Bipartite graph and Fan graph of any size by suitably assigning values to the variable m.

Example 4.1. Bipartite Graph $K_{m,n}$: Let us consider an initial hypergraph G_0 and construct the rules of the grammar P_1 and P_2 as given in the following figure. In this example we have shown that any bipartite graph $K_{m,n}$ can be generated using the grammar R by giving suitable values to the variable m.

G_0 : 

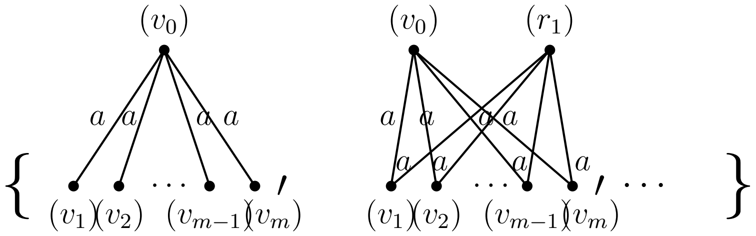


Derivation :



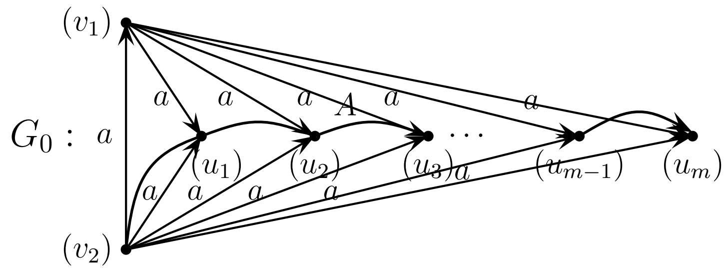
The graph language generated by the grammar R is the family of Bipartite graphs $K_{m,n}$ and be given in the following figure:

$R^\star(H) =$

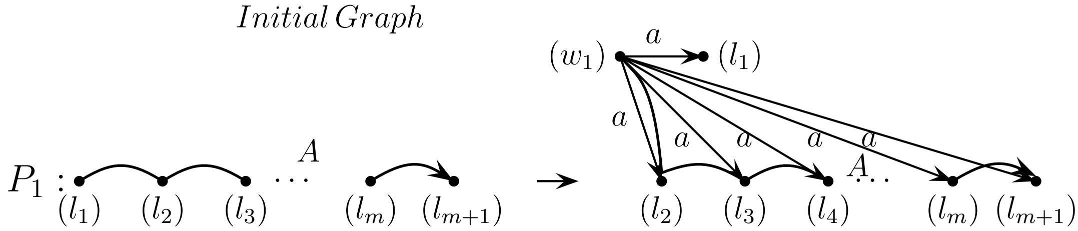


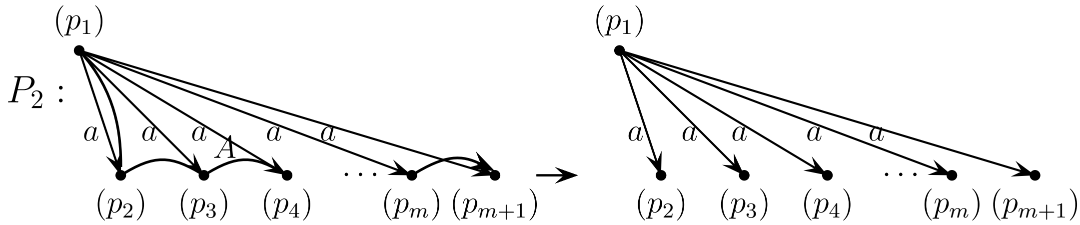
Example 4.2. Fan Graph $F_{\{m,n\}}$: The fan graph $F_{\{m,n\}}$ defined as the graph joined $\bar{K}_m + P_n$, where $\bar{K}_m$ is the empty graph on m nodes and P_n is the path graph on n nodes.

Let us consider an initial hypergraph G_0 and construct the rules of the grammar R as shown below. In this example we have shown that any $F_{\{m,n\}}$ graph can be generated by the grammar R by giving suitable values to the variable m.

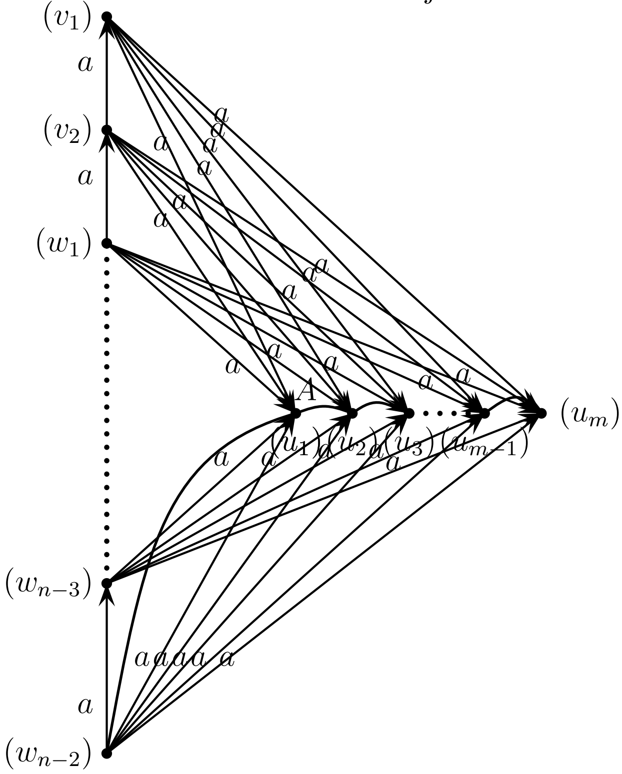


Initial Graph

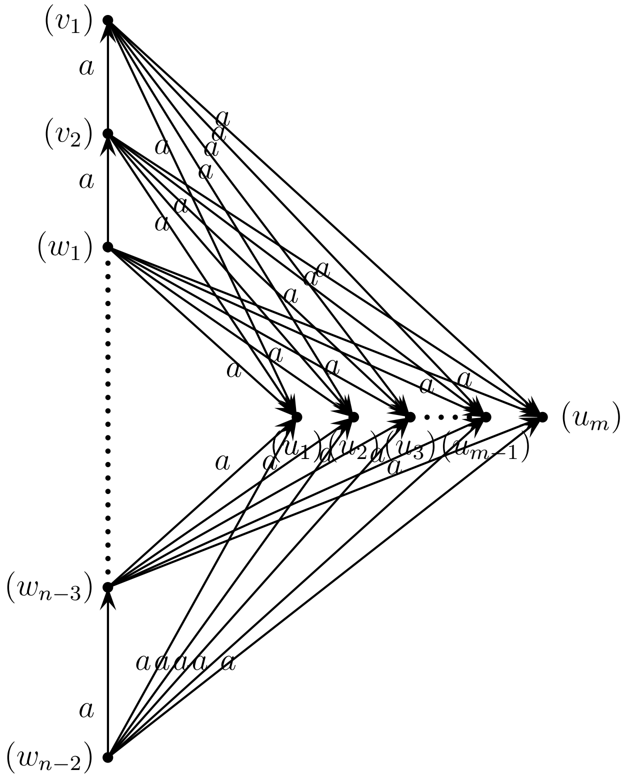




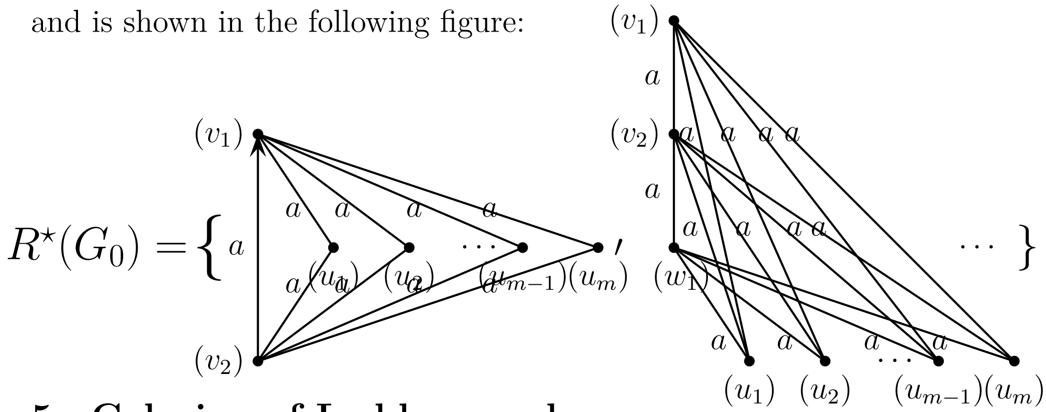
Starting with an initial graph G_0, applying the production P_1, $(n-2)$ times we get a graph as shown in the following figure.



Then applying the production P_2 we get the fan graph $F_{\{2,n\}}$ as shown in the following figure.



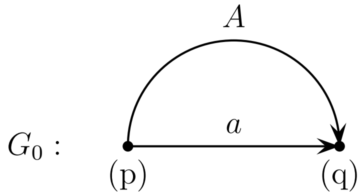
The graph language of the above grammar is the family of Fan graphs $F_{\{m,n\}}, m \geq 2$ and is shown in the following figure:



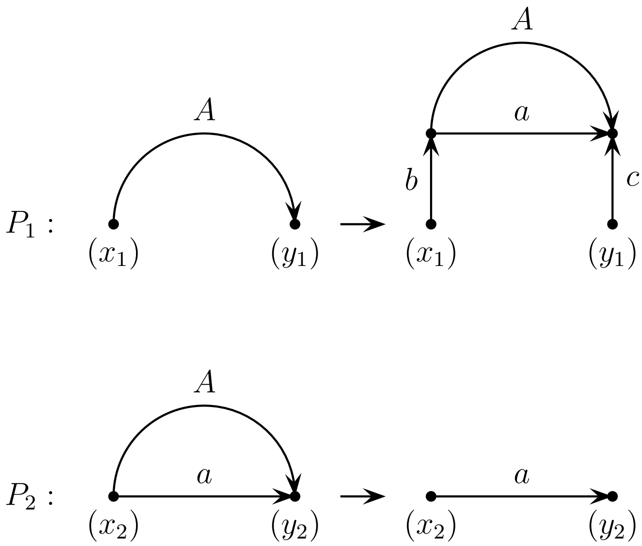
5 Coloring of Ladder graph

This section deals with coloring of Ladder and Friendship graph. Though the graphs are simple and easily colorable, we have followed new approach in coloring them. We have adapted hypergraph grammar as a tool in our approach.

In the mathematical field of graph theory, the ladder graph L_n is a planar undirected graph with $2n$ vertices and $\big(n+2(n-1)\big)$ edges. The ladder graph can be obtained as the cartesian product of two path graphs, one of which has only one edge: $L_{n,1} = P_n \times P_1$.

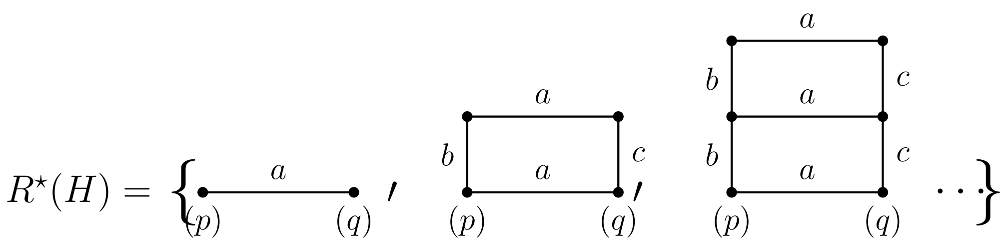


Initial Graph

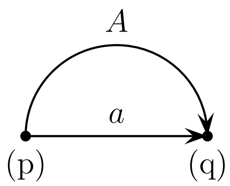
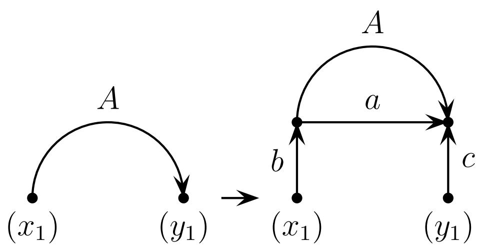


Productions

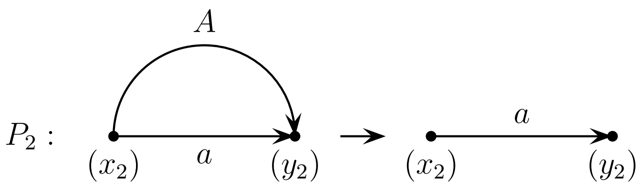
Starting with the base graph G_0, by applying the rule P_1 repeatedly $(n-1)$ times and then P_2 once we get a ladder graph with $2n$ vertices where $n \geq 1$. The language generated by the grammar R is the family of ladder graphs with $2n$ vertices $(n \geq 1)$ and is shown in the following figure.



Implementation of Syntax Directed Translators for coloring: Syntax - directed translation scheme is a convenient description of what we would like to be done. A syntax directed translation scheme is merely a grammar in which a semantic action is associated with each production. Whenever a rule is used in the derivation, the action is taken. In this section, we have written semantic action for coloring graphs. The following figure shows the abstract translation scheme.

Production	**Semantic Action**
G_0 : 	$p.col = color1$ $q.col = color2$
P_1 : 	$x_1.col = color\Big(X_{i-1}(2)\Big)$ $y_1.col = color\Big(X_{i-1}(3)\Big)$ $Q(2).col = y_1.col$ $Q(3).col = x_1.col$

$x.col$ and $y.col$ are translations whose value is a color. Let X_{i-1} be the non-terminal hyperarc which is existing in G_{i-1}. The function $color\Big(X_{i-1}(j)\Big)$ is the color of the vertex of X_{i-1} which is in the j^{th} position. This function takes two arguments. Let Q be a



$x_2.col = X_{n-1}(2)$
$y_2.col = X_{n-1}(3)$

labelled hyperarc existing in $Im(P_1)$. $Q(i).col$ is the color of the vertex which is in the i^{th} position. $Q(2).col$ and $Q(3).col$ are the colors of the vertices which are the head and tail of Q respectively.

Let G_k be a hypergraph generated from G_0 in the k^{th} step of parallel deviation and X_{k-1} be a non-terminal hyperarc existing in G_{k-1}. According to the rewriting principle, we can say that

$$G_k = \begin{cases} \sqsubset G_{k-1} \sqsupset +h_k(H_1), & 1 \leq k \leq (n-1) \\ \sqsubset G_{k-1} \sqsupset +h_k(H_2), & k = n \end{cases}$$

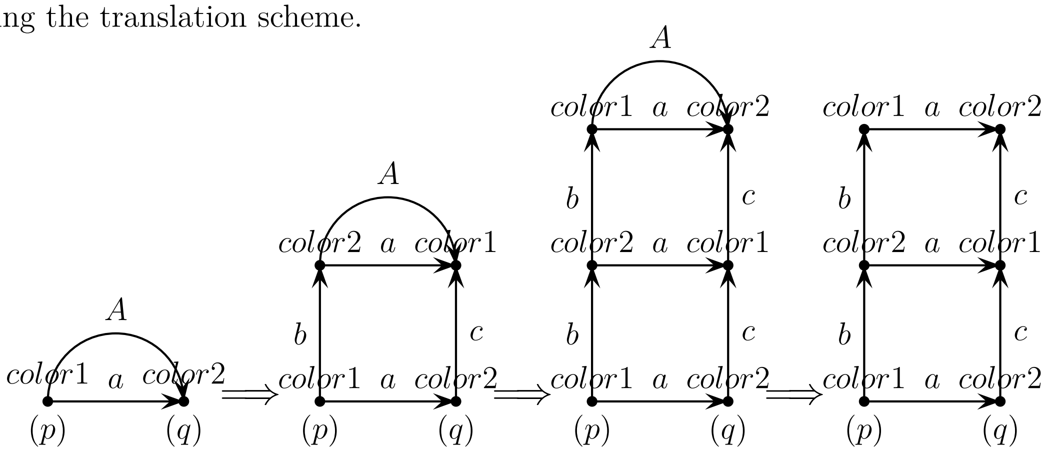
where H_1 and H_2 are $Im(P_1)$ and $Im(P_2)$ respectively. Obviously, one color is not sufficient to color G_k properly because $\sqsubset G_0 \sqsupset$ itself need two colors. Hence $\chi(G_k) \neq 1$. We can prove that G_k is 2-colorable. For $k = 0$, we need two colors to color G_0 properly as there is an adjacency between the vertices p and q. By induction, we assume that G_{k-1} is 2-colorable. we have to prove that G_k is 2-colorable. The graph G_k is obtained from G_0, by applying either P_1 or P_2.

Case 1 : While applying P_1. Under the transformation h_k, the vertices x_1 and y_1 are transformed to $X_{k-1}(2)$ and $X_{k-1}(3)$ respectively and so G_k has two more vertices in addition to the vertices of G_{k-1}. By the coloring principle, the input vertices x_1 and y_1 are colored by the color of the vertices $X_{k-1}(2)$ and $X_{k-1}(3)$ respectively. Therefore, the color of the vertices $X_{k-1}(2)$ and $X_{k-1}(3)$ in G_k is as same as in G_{k-1} and the colors of them are color 1 and color 2 respectively. From the above, we are in need to verify the coloring of $h_k(H_1)$ and not $\sqsubset G_{k-1} \sqsupset$. By the coloring principle, the vertices $Q(2)$ and $Q(3)$ receive the color of the vertices y_1 and x_1 respectively and so $h_k(Q(2)$ and $h_k(Q(3)$ receive the color of the vertices y_1 and x_1 respectively. The vertices adjacent to $h_k(x_1)$ and $h_k(y_1)$ are $h_k\Big(Q(2)\Big)$ and $h_k\Big(Q(3)\Big)$ respectively. As the color of the vertices x_1 and y_1 are being assigned to $Q(3)$ and $Q(2)$ respectively, the end vertices of the three new edges which are existing in G_k between the pair of vertices such as $\Big(h_k(x_1), h_k\big(Q(2)\big)\Big)$, $\Big(h_k(y_1), h_k\big(Q(3)\big)\Big)$ and $\Big(h_k\big(Q(2)\big), h_k\big(Q(3)\big)\Big)$ receive distinct colors.

Case 2 : While applying P_2. The input vertices x_2 and y_2 are transformed to the vertices namely $X_{n-1}(2)$ and $X_{n-1}(3)$ respectively under the transformation h_n. The vertex x_2 also receives the color of the vertex $X_{n-1}(2)$ and the vertex y_2 receives the color of the vertex $X_{n-1}(3)$. Therefore, the color of the vertices $X_{n-1}(2)$ and $X_{n-1}(3)$ in G_n as same as in G_{n-1}. The end vertices of an edge $\big(h_n(x_2), h_n(y_2)\big)$ are distinct in color as the color of the vertices $X_{n-1}(2)$ and $X_{n-1}(3)$ be distinct. Hence, G_k is 2 colorable $\forall\, k = 1, 2, \cdots n$.

Trace of syntax directed Translation:

If we derive the Ladder graph with 6 vertices, we get the following colored ladder graph using the translation scheme.

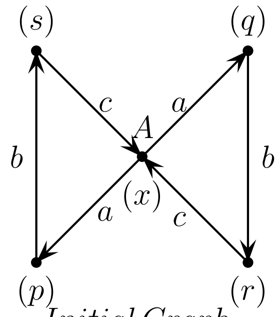


6 Coloring of Friendship graph

In the mathematical field of graph theory, the friendship graph (or Dutch windmill graph or n-fan) F_n is a planar undirected graph with 2n+1 vertices and 3n edges.The friendship graph F_n can be constructed by joining n copies of the cycle graph C_3 with a common vertex.

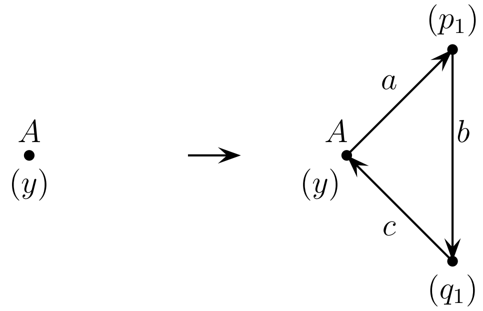
Let us consider the grammar $R = (G_0, P)$ where G_0 is an initial graph and $P = \big\{P_1, P_2\big\}$ is the set of rules to generate a Friendship graph $F_n (n \geq 2)$.

G_0 :

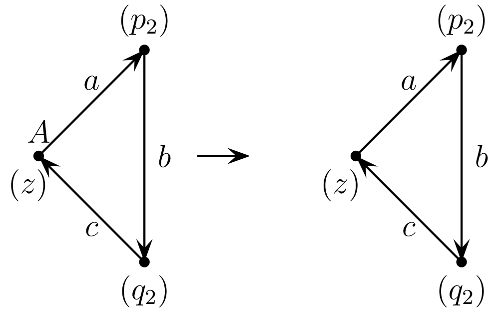


Initial Graph

P_1 :

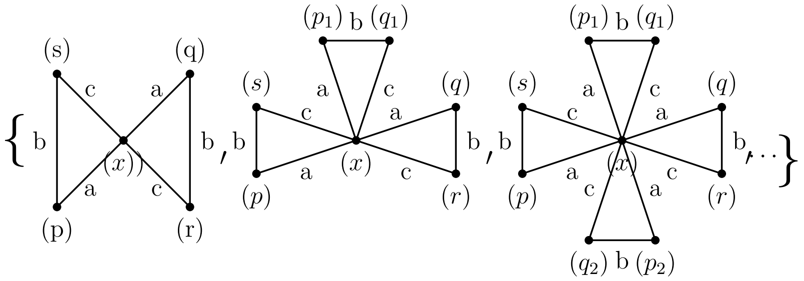


P_2 :



Starting with an initial graph G_0, applying the rule P_1 repeatedly $(n-2)$ times and then applying P_2 once we get the friendship graph $F_n, (n \geq 2)$. The language generated by the Grammar G is the family of friendship graphs $F_n \ \ (n \geq 2)$.

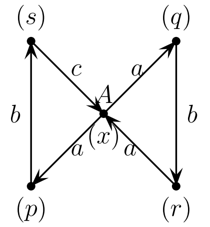
$R^\star(G_0) :=$



We can use the principle of syntax directed translation scheme in the productions of the grammar so as to get coloring on the friendship graph. The syntax directed scheme for the above grammar be given as follows:

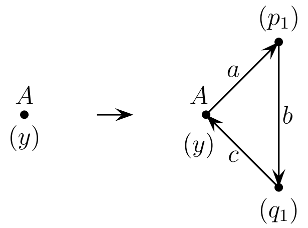
Production *Semantic rule*

G_0 :



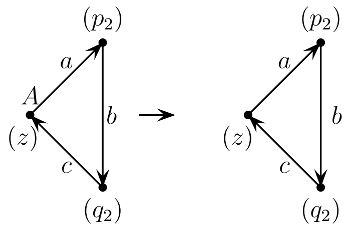
$\{$
$p.col = color1$
$x.col = color2$
$s.col = color3$
$q.col = p.col$
$r.col = s.col$
$\}$

P_1 :



$\{$
$y.col = x.col$
$p_1.col = p.col$
$q_1.col = s.col$
$\}$

P_2 :



$\{$
$z.col = x.col$
$p_2.col = p.col$
$q_2.col = s.col$
$\}$

$p.col$, $x.col$, and $s.col$ are translations whose value is a color.

Let G_k be a hyper graph generated from G_0 in the k^{th} step of parallel deviation and X_{k-1} be a non-terminal hyperarc existing in G_{k-1}. According to the rewriting principle, we can say that

$$G_k = \begin{cases} \sqsubset G_{k-1} \sqsupset +h_k(H_1), & 1 \leq k \leq (n-2) \\ \sqsubset G_{k-1} \sqsupset +h_k(H_2), & k = n-1 \end{cases}$$

where H_1 and H_2 are $Im(P_1)$ and $Im(P_2)$ respectively. Obviously, two colors are not sufficient to color G_k properly because $\sqsubset G_0 \sqsupset$ itself need three colors. Therefore, $\chi(G_k) \neq 2$ and hence $\chi(G_k) \neq 1$. We can prove that G_k is 3-colorable.

For $k = 0$, we need three colors to color G_0 properly as there is an edge between each pair of vertices (x,p), (p,s) and (x,s) labelled a. The vertex q can be colored by the color of the vertex p as there is no adjacency between p and q. In the same way, the vertex r can be colored by the color of the vertex s. Thus, G_0 is 3 colorable. By induction, we assume that G_{k-1} is 3-colorable. we have to prove that G_k is 3-colorable. The graph G_k is obtained from G_0 by applying either P_1 or P_2.

Case 1 : While applying P_1. Under the transformation h_k, the vertex y is transformed to the vertex x which is existing in G_{k-1} and so G_k has two more vertices in addition to the vertices of G_{k-1}. By the coloring principle, we color the input vertex y by the color of the vertex x and so the color of the vertex x in G_k be as same as in G_{k-1} and it is color2. Therefore, we are in need to verify the coloring of $h_k(H_1)$ but not $\sqsubset G_{k-1} \sqsupset$.

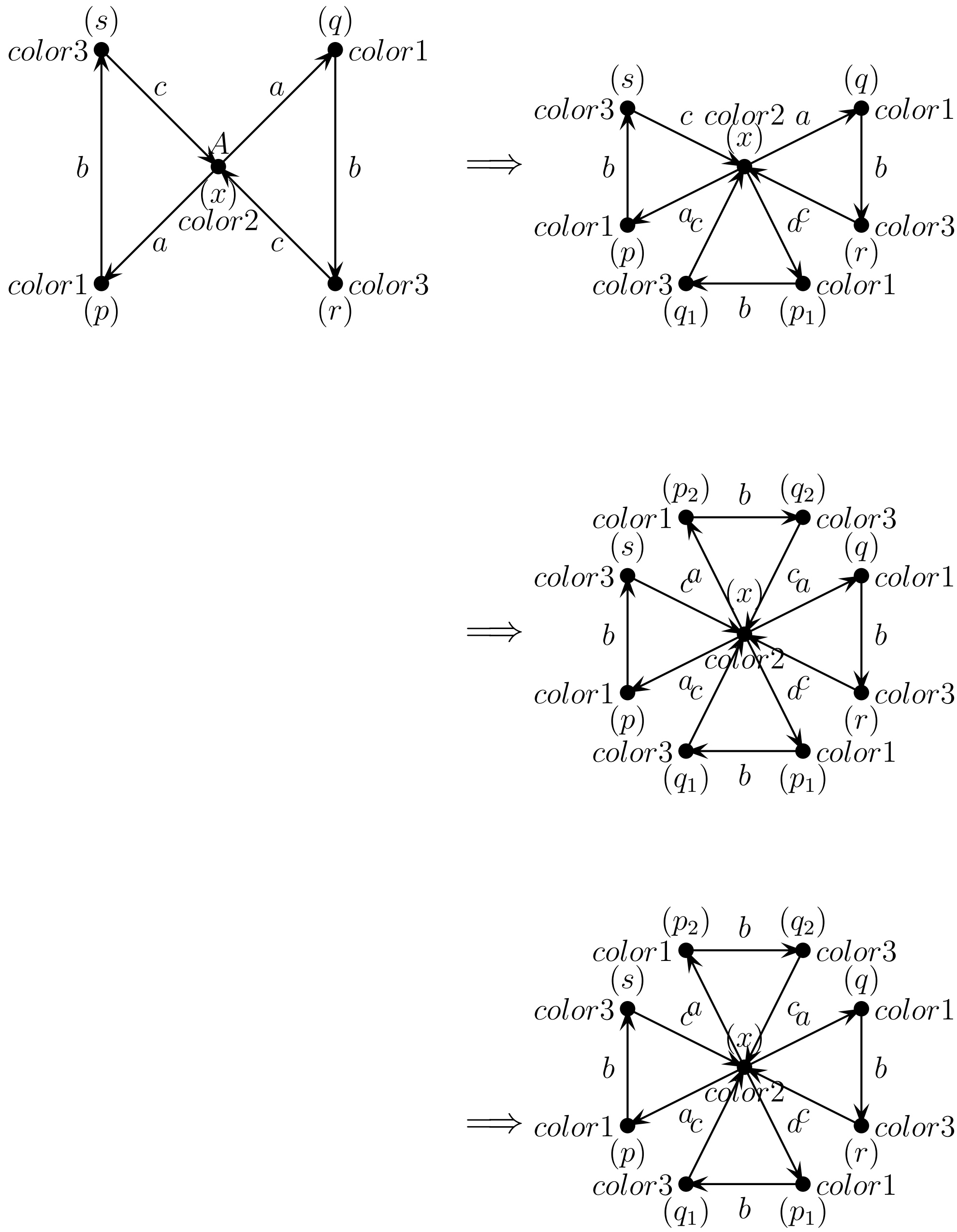
The vertices p_1 and q_1 of $Im(P_1)$ receive the color of the vertices p and s respectively and so $h_k(p_1) = p_k$ and $h_k(q_1) = q_k$ are colored in G_k by the colors color1 and color3 respectively. The vertices adjacent to $h_k(y)$ are $h_k(p_1)$ and $h_k\Big(Q(3)\Big)$. As the color of the vertices p and s are assigned to p_1 and q_1 respectively, the end vertices of the three new edges which are existing in G_k between the pair of vertices such as $\Big(h_k(y), h_k(p_1)\Big)$, $\Big(h_k(q_1), h_k(y)\Big)$ and $\Big(h_k(p_1), h_k(q_1)\Big)$ receive distinct colors.

Case 2 : While applying P_2**.**

Under the transformation h_{n-1}, the input vertices z, p_2 and q_2 are transformed to the vertices of G_{n-2} namely x, p_{n-2} and q_{n-2} respectively. Hence, the vertices of G_{n-1} are the vertices of G_{n-2}. As the input vertex z is colored by the color of the vertex x, the color of the vertex x in G_{n-1} is as same as in G_{n-2} and it is color2. The vertices p_2 and q_2 of Im (P_2) receive the color of the vertices p and s respectively as p_1 and q_1 of P_1. Therefore, the color of the vertices $h_{n-1}(p_2) = p_{n-2}$ and $h_{n-1}(q_2) = q_{n-2}$ in G_{n-1} will be as same in G_{n-2} and so the end vertices of the edges $\Big(h_{n-1}(z)\ ,h_{n-1}(p_2)\Big)$, $\Big(h_{n-1}(p_2)$, $h_{n-1}(q_2)\Big)$ and $\Big(h_{n-1}(p_2), h_{n-1}(z)\Big)$ receives distinct colors. Thus, G_{n-1} is 3 colorable.

Trace of syntax directed Translation:

If we derive the Friendship graph F_4, we get the following colored Friendship graph using the translation scheme.



7 Conclusion

The author has introduced the concept of regular multi- hypergraph and multi-regular hypergraph language which allows multiplicity on the occurrence of terminal hyperarcs. We have used hypergraph grammar as a tool to generate some family of graphs that are special in Graph theory such as Ladder graph and Friendship graph. The author has proposed an algorithm using syntax directed translation scheme to produce coloring.

References

[1] Caucal .D, Handbook of Deterministic Graph Grammars, University of Paris-Est.

[2] Caucal .D, On the regular structure of prefix rewriting, 15th CAAP, LNCS 431, A. Arnold (Ed.), 87–102 1990, and in Theoretical Computer Science 106,61-86 (1992).

[3] Frank Harary, Graph Thoery, Narosa Addison Wesley, Indian Student Edition,1988.

[4] Rozenberg .G, editor. Handbook of Graph Grammars and Computing by Graph Transformation, Volume 1-3: Foundations. World Scientific, 1997.