数理解析研究所講究録
第625巻 1987年 55-63

Graph grammars with path controlled embedding

by

Kunio Aizawa （会沢　邦夫）
and
Akira Nakamura （中村　昭）

Department of Applied Mathematics
Hiroshima University
Higashi-Hiroshima, 724
Japan

1. Introduction

In the recent years, many models for graph grammars have been proposed as a kind of hyper-dimensional generating systems (see e.g., [1]-[4]). The basic primitive in a graph grammar is a directed (or non-directed) graph which is a set of labelled vertices and labelled edges. Thus, most of the graph grammars have some kind of the "embedding" rules as well as the "production" rules to embed the newly replaced graph to its host graph. On the other hand, many models for two-dimensional formal grammars have also been proposed. These grammars generate two-dimensional patterns embedding in two-dimensional arrays. Generally, in those formal grammars, to preserve topological relations between each point in generated arrays a restriction called "isometric" is inserted in each production rule.

One of the reasons why both of above mentioned generating systems were proposed is to generalize the (one-dimensional or string) formal grammars to two-dimensional or more complex structures. However, few has known about the relationships between graph grammars and two-dimansional formal grammars. It is obvious that the graph grammars with no restrictions are more powerful than the two-dimensional formal grammars since there exist many graphs which cannot be embedded in any two-dimensional structures. Thus, in order to investigate such relationships, the graph grammars are restricted such that they can generate planer graphs only. In the two-dimensional formal grammars, the "array" is used as an implicit structure of the generated patterns (i.e.,two-dimensional words). Even if restricted to planer graphs, almost all structures, on the other hand, can be formed in the graph grammars by making use of various "embeddeing" rules. Therefore, it is necessary to restrict the array grammars within the graphs having

the two-dimensional array structure. It has been shown in [5], however, that such restriction may bring sufficient generative power as a isometric array grammar into a graph grammar having production rules of the context-free style.

In this paper, we will define graph grammars called node-replacement graph grammars with path controlled embedding (nPCE grammars) which use a sequence of edges instead of the single edge to embedding a newly replaced graph into the host graph. Then we will show that there exists a class of nPCE grammars generating a class of graph languages which is equal to the class of context-free aray languages under certain stipulation. It means that the path controlled embeding brings no significant contexts into the production rules. Then, we will characterize the above mentioned class of nPCE grammars by making use of the partial path group firstly introduced in [6].

We assume for readers to be familiar with the theories of two-dimensional grammars and graph grammars (see, e.g., [7] and [8]).

3. Basic definitions

In this section, we will define nPCE graph grammars and their languages. We also review here the definitions of the partial path groups. For more precise definitions of the partial path groups, see [6] and [9].

Definition 1. Let H be a connected graph of degree d, i.e., no more than d edges emanate from any node. By an edge coloring of H we mean an assignment of colors to the edges of H such that the edges emanating from any given node all have different colors.

Definition 2. Let p be a node of a graph H and let π be a string of colors. Then $p\pi$ is defined as the terminal node of the path defined by π starting from p, provided this path is realizable. For convenience, let us define a fictitious "blank" color representing "no move". It is obvious that the structure defined above resembles a group structure called "partial group". From now on we shall refer to this partial group as the partial path group of H defined by given coloring, and denote it by $\Pi(H)$.

Definition 3. A partial path group $\Pi(H)$ is called near-abelian if each color except blank color commutes with all but one of the other colors, and does not commute with the remaining one.

Note that the partial path groups can be defined on a graph generated by a

graph grammar by regarding the edge labels of the generated graph to the colors mentioned in above definitions.

Definition 4. A node-replacement graph grammar with path controlled embedding, denoted as nPCE grammar, is a construction $G = \langle \Sigma_N, \Sigma_E, P, Z, \Delta_N, \Delta_E \rangle$, where

Σ_N is a finite nonempty set of node labels,

Σ_E is a finite nonempty set of edge labels,

Δ_N is a finite nonempty subset of Σ_N, called terminal node labels,

Δ_E is a finite nonempty subset of Σ_E, called terminal edge labels,

P is a finite set of productions of form (a, β, ψ), where a is a node, $\beta = (V, E, \Sigma_V, \Sigma_E, \phi_V, \phi_E)$ is a connected graph and ψ is a mapping from Σ_E^+ into $2^{(V(\beta) \times \Sigma_E)}$; ψ is called embedding function.

Z is a connected graph over (Σ_N, Σ_E) called the axiom.

A directed derivation step in a nPCE grammar is performed as follows:

Let $H = (V, E, \Sigma_V, \Sigma_E, \phi_V, \phi_E)$ be a graph. Let $p = (a, \beta, \psi)$ be a production in P. Let β' be isomorphic to β (with h an isomorphism from β' into β), where β' and H-a have no common nodes. Then the result of the application of p to H (by using h) is obtained by first removing a from H, then replacing a with β' and finally adding edges (u,v) between every nodes u in β' and every v in H-a such that there exists a path $\pi \in \mathrm{dom}(\psi)$ with $v = a\pi$ in H provided that:

(1) let c be an edge label, let v be a node of a graph H and let π, π' be strings of edge labels. Let $c\pi$ and $c\pi'$ be in the domain of a ψ. If both $v(c\pi)$ and $v(c\pi')$ are realizable on H and $v(c\pi) = v(c\pi')$, then only one of these strings is selected (nondeterministically) to embed the replaced graph.

(2) Let v be a node of a graph H, let π, π' be strings of edge labels and let their prefix symbols do not equal. For the case of $v\pi = v\pi'$, the generation of G can be proceed if and only if $\psi(\pi) = \psi(\pi')$.

Note here that the embedding function ψ bring no context into the generation procedure of an nPCE grammar since no node labels are refered in any place of embedding steps except the one of the newly replaced graph.

Formally the notation of a direct derivation step is defined as follows.

Definition 5. Let $G = \langle \Sigma_N, \Sigma_E, P, Z, \Delta_N, \Delta_E \rangle$ be a nPCE grammar and let H, H'be graphs over (Σ_V, Σ_E).

(1) H directly derives H' in G, denoted as $H \Rightarrow_G H'$, if there exists a production

$p = (a, \beta, \psi)$ in P, a graph β' with $V(\beta') \cap V(H-a) = \emptyset$ and an isomorphism h from β' into β such that H' is isomorpic to the graph X constructed as follows:

$X = (V, E, \Sigma_V, \Sigma_E, \phi_V, \phi_E)$, where

$V = V(H-a) \cup V(\beta')$,

$E = \{ (x,y) \mid x,y \in V(H-a) \text{ and } (x,y) \in E(H) \}$

$\cup \{ (x,y) \mid x,y \in V(\beta') \text{ and } (h(x),h(y)) \in E(\beta) \}$

$\cup \{ (x,y) \mid x \in V(\beta'),\ y \in V(H-a),$ there exists a path $\pi \in dom(\psi)$ with $y = a\pi$ and $\langle h(x), e \rangle \in \psi(\pi)$ for some e in $\Sigma_E \}$,

ϕ_V is equal to the node labelling function of H for nodes in V(H-a), equal to the node labelling function of β' for nodes in $V(\beta')$,

and ϕ_E is equal to the edge labelling function of H for edges between the nodes of H, is equal to the edge labelling function of β' for edges between the nodes of β', is defined by ψ for the edges between the node of H and of β'.

We also say that H' is derived from H by replacing a using the production p.

(2) We will denote the reflexive and the transitive closure of $\Rightarrow_G$ by $\Rightarrow_G^*$ and the transitive closure of $\Rightarrow_G$ by $\Rightarrow_G^+$.

(3) The language of G, denoted as L(G), is defined by $L(G) = \{ H \mid H$ is a graph over (Σ_V, Σ_E) and $Z \Rightarrow_G^* H \}$.

Example 1. The following nPCE grammar G generates the set of all "rectangular grids" of the form represented in Fig. 1:

$G = \langle \Sigma_N, \Sigma_E, P, Z, \Delta_N, \Delta_E \rangle$, where

$\Sigma_N = \{ A, a \}$, $\Sigma_E = \{ h, h', v, v', D \}$, $\Delta_N = \{ a \}$, $\Delta_E = \{ h, h', v, v' \}$, $Z =$ (A) ,

and P = { ((A)$_1$, (A)$_1$ —r— (A)$_2$, $\psi(r') = (1, r')$

$\psi(s) = (1, s)$

$\psi(s') = (1, s')$

$\psi(sr) = (2, s)$

$\psi(s'r) = (2, s')$

$\psi(r) = (2, D)$

$\psi(srs) = (2, D)$

$\psi(s'rs') = (2, D)$

$\psi(srr'sr') = (2, D)$

$\psi(s'rr's'r') = (2, D)$

$\psi(srr'ss'r's') = (2, D)$

$\psi(s'rr's'sr's) = (2, D)$

for every pair r = h, r' = h', s = v, s = v'

or r = h, r' = h', s = v', s = v
or r = h', r' = h, s = v', s = v
or r = h', r' = h, s = v, s = v'
or r = v, r' = v', s = h, s = h'
or r = v, r' = v', s = h', s = h
or r = v', r' = v, s = h', s = h
or r = v', r' = v, s = h, s = h'),

((A, 1), (a, 1), ψ(h) = (1, D), ψ(h') = (1, D),
ψ(v) = (1, D), ψ(v') = (1, D),
ψ(vvh) = (1, D), ψ(vvh') = (1, D),
ψ(v'v'h) = (1, D), ψ(v'v'h') = (1, D),
ψ(hhv) = (1, D), ψ(hhv') = (1, D),
ψ(h'h'v) = (1, D), ψ(h'h'v') = (1, D),
ψ(vvhhv) = (1, D), ψ(vvh'h'v) = (1, D),
ψ(v'v'hhv') = (1, D), ψ(v'v'h'h'v') = (1, D),
ψ(hhvvh) = (1, D), ψ(hhv'v'h) = (1, D),
ψ(h'h'vvh') = (1, D), ψ(h'h'v'v'h') = (1, D),
ψ(v'hv'hh) = (1, h'), ψ(vhvhh) = (1, h),
ψ(hvhvv) = (1, v), ψ(hv'hv'v') = (1, v'),
ψ(v'hv'hh) = (1, h), ψ(v'h'v'h'h') = (1, h'),
ψ(h'v'h'v'v') = (1, v'), ψ(h'vh'vv) = (1, v),
ψ(vh'vh'h'vh') = (1, v), ψ(vhvhhvh) = (1, v),
ψ(hvhvvhv) = (1, h), ψ(hv'hv'v'hv') = (1, h),
ψ(v'hv'hhv'h) = (1, v'), ψ(v'h'v'h'h'v'h') = (1, v'),
ψ(h'v'h'v'v'h'v') = (1, h'), ψ(h'vh'vvh'v) = (1, h')) } .

Each time when a new node is added to the host graph by using the first production rule, the attached embeddeing function ψ ensure the local connection around the node, i.e., neighborhood relations of the newly added node. An example of a derivation of this step is given in Fig. 2. Finally, for each node, G checks whether it and its 8-neighbors (some may be ommitted) form a rectangular by using the embedding function of the second production rule. The function utilize nine paths given in Fig. 3 for each neighbor. It is not so difficult to see that if each node and its 8-neighbors form a rectangular, then the generted graphs shape a rectangular form.

4. nPCE grammars having the 2D array structure

In this section, we will characterize a class of nPCE grammars which generate the set of all 2D rectangular grids by making use of partial path group. Then, we will

prove that there exists a class of nPCE grammar generating a class of graphs which is regarded as the class of context-free array languages under some stipulations.

Stipulation 1. (1) A symbol in an array is regarded as a label of the corresponding node of a graph.

(2) A horizontal connection of two points is regarded as an edge labelled with h or h'. These two labels appear alternatively along a row.

(3) A vertical connection of two points is regarded as an edge labelled with v or v'. These two labels appear alternatively along a column.

Let L be a set of arrays. Let L' be a set of graphs such that each element α of L' there exists one and only one element of L which is regarded as the graph α in the sense of Stipuration 1. Then we will use a notation $L =_1 L'$ to represent above relation.

Stipulation 2. An array is regarded as a graph, which takes the square pattern, such that if all nodes labelled with #s are ignored, the array is regarded as the graph in the sense of Stipulation 1.

As in the case of Stipulation 1, we will use a notation $L =_2 L'$.

Lemma 1. For every CFAG T there exists an nPCE grammar G such that $L(G) =_2 L(T)$.

Proof: Let $T = \langle \Sigma, \Delta, \#, P, S \rangle$ and let $P = \{ p_1, p_2, \ldots, p_n \}$. Without loss of generality, we can assume that T has the normal form, i.e., the right hand side of each production rule have at most two nonterminal symbols or one terminal symbol. Then we construct the following nPCE grammar:

$$G = \langle \Sigma_N, \Sigma_E, \wp, Z, \Delta_N, \Delta_E \rangle,$$

where $\Sigma_N = \Sigma \cup \Delta' \cup \{ \# \}$, $\Sigma_E = \{ h, h', v, v', D \}$, $\Delta_N = \Delta \cup \{ \# \}$, $\Delta_E = \{ h, h', v, v' \}$, $\wp$ is defined in Appendix and Z is a grph having only one node labelled with S. Here we have that $\Delta' = \{ x' \mid x \in \Delta \}$.

The following outline of generation steps of G will be useful in order to understand this lemma.

(1) Starting with the graph Z, G simulates the derivation of T by using the production rules of type 1. In general, each single step of T is corresponding to the single step of G. For each node labelled with a synbol from Σ, G replace the node with a graph by applying a production rule of type 1. Then, the newly replaced graph is embedded by making use of the attaced embedding function. If there are

sufficient space to embed such replaced graph, then each node of embedded graph is connected with appropriate nodes of host graph by the edges labelled with the symbols from Δ_E. If not, a node of embedded graph is connected by at least one edges labelled with the "dead" label D. The edges labelled with D cannot be replaced by any other edges. Note here that at this point, such embedding ensure only local connection of the embedded graph. This procedure is repeated to simurate the generating steps of T.

(2) At any steps of above procedure, G can create the nodes labelled with #, the blank symbol of T, to form a rectangular. In this step, the embedding procedure is same as the one of the production rules of type 1.

(3) Finally, G must ensure the global connections of the generated graph. To do this, G uses the production rules of type 3. For every nodes labelled with the symbols from Δ', these production rules ensure that each node and its 8-neighbors (some may be ommitted) form a rectangular by using almost same way to the nPCE grammar in Example 1. Note here that if a inconsistency occurs when two nodes are connected, the generating procedure stops and no terminal graphs are generated. It is not so difficult to see that if each nodes are connected correctly with their neighbors and the generated graph has the rectangular form, then global connections of each nodes in the generated graph is correct.

In this way, we obtain $L(G) =_2 L(T)$.

In the above construction of the production of G, we can use the free near-abelian partial path groups to define the domain of the embedding function ψ. In this case, arbitrarily long paths can be used to embed the newly replaced graphs. Thus, G can ensure the global correctness of connections at any time. So it is no longer necessary that the generated graphs have rectangular form. Therefore we get the next lemma immediately.

Lemma 2. For every CFAG T there exists an nPCE grammar G characterized with the free near-abelian partial path groups such that $L(G) =_1 L(T)$.

It is shown in [6] that the free near-abelian partial path groups having four colors correspond to a subgraph of two-dimensional array. From the fact and Lemma 2, it is not so difficult to see the next teorem.

Theorem 1. There exists a class of nPCE grammars characterized with the free near-abelian patial path group which is equal to the class of two-dimensional

context-free array grammars under Stipulation1.

In above Theorem, if it is required to use the paths having only fixed finite length, then it is necessary to use clumsy production rules discussed in the proof of Lemma 1. In this case, Stipulation 2 must be used instead of Stipulation 1.

5. Concluding Remarks

In this paper, we have intoroduced a node-replacement graph grammars called nPCE grammars which use paths of edges to embed the replaced graph. We have shown that there exists a class of nPCE grammars which is equal to the class of two-dimensional context-free array grammars. Such class of nPCE grammars can be characterized by the free near-aberian partial path groups.

In [6] and [9], some relationships between the partial path groups and the two-dimansional patterns. So it is not difficult to construct and characterize nPCE grammars generating various two-dimensional patterns such as the triangular array, hexagonal array and so on.

References

[1] Janssens, D., and G. Rozenberg: On the structure of node label controlled graph languages, Information Sciences, 20, pp.191-216 (1980).

[2] Janssens, D., and G. Rozenberg: Restrictions, extensions, and variations of NLC grammars, Information Sciences, 20, pp.217-244 (1980).

[3] Janssens, D., and G. Rozenberg: Graph grammars with neighborhood-controlled embedding, Theoretical Computer Science, 21, pp.55-74 (1982).

[4] von Solms, S.H.,: Node-label controlled graph grammars with context conditions, International Journal of Computer Mathematics, 15, pp.39-49 (1984).

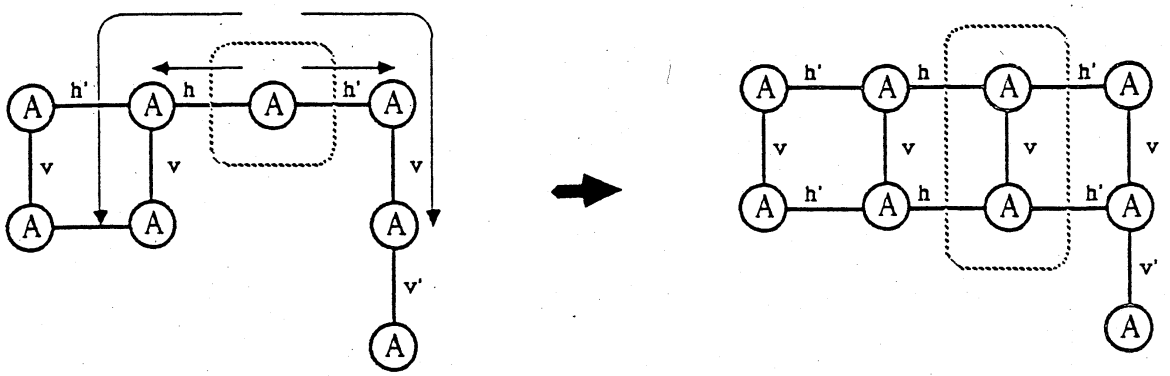
[5] Nakamura, A., and K. Aizawa: On a relationship between graph L-systems and picture languages, Theoretical Computer Science, 24, pp.161-177 (1983).

[6] Rosenfeld, A.,: Partial path groups and parallel graph contractions, in G. Rozenberg and A. Salomaa (eds.), The Book of L, Springer-Verlag, pp.369-382 (1986).

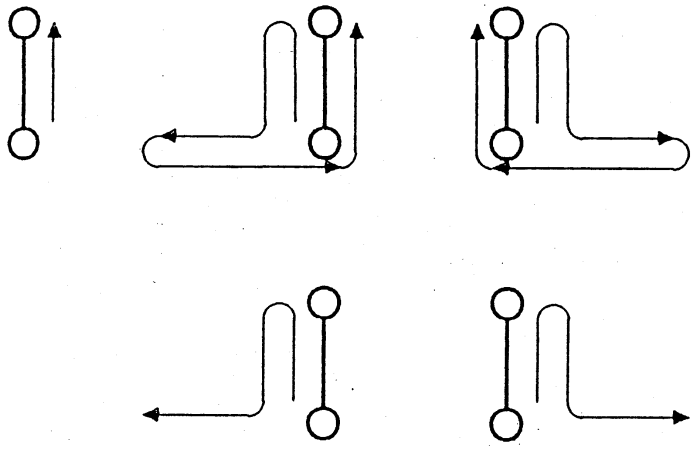
[7] Rosenfeld, A.,: Picture Languages, Academic Press (1979).

[8] Nagl, M.,: A tutorial and bibliographcal survey on graph grammars, Lecture Notes in Computer Science, 73, Springer-Verlag (1979).

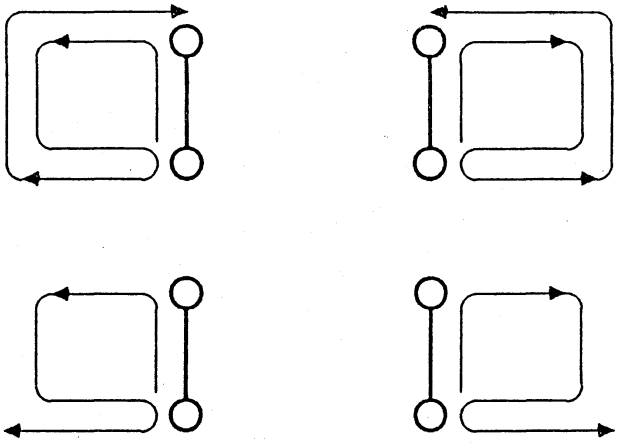
[9] Melter, R. A.,: Tessellation graph characterization using rosettas, Pattern Recognition Letters, 4, pp.79-85 (1986).



Using Paths to embed a doughter graph into the host graph.



(a) Cases of connected by edges labelled with the "dead symbol".



(b) Cases of connected by edges labelled with the normal labels.