<table>
<thead>
<tr>
<th>Title</th>
<th>Syntactic Characterization of Two-Dimensional Grid Graphs by a Context-Sensitive Graph Grammar (New Trends in Theory of Computation and Algorithm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>ARITA, Tomokazu; TSUCHIDA, Kensei; YAKU, Takeo</td>
</tr>
<tr>
<td>Citation</td>
<td>数理解析研究所講究録 1489: 78-84</td>
</tr>
<tr>
<td>Issue Date</td>
<td>2006-05</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/2433/58226">http://hdl.handle.net/2433/58226</a></td>
</tr>
<tr>
<td>Type</td>
<td>Departmental Bulletin Paper</td>
</tr>
<tr>
<td>Textversion</td>
<td>publisher</td>
</tr>
</tbody>
</table>
Syntactic Characterization of Two-Dimensional Grid Graphs by a Context-Sensitive Graph Grammar *

Tomoaki Arita (Tomokazu ARITA)
Department of Languages and Information Studies, Obirin University

Kensei Tsuchida (Takeo YAKU)
Department of Information and Computer Sciences, Toyo University

Abstract

Graph grammar can characterize several type of graphs. Vigna and Ghezzi showed that the language of grid graphs could not be constructed by their context-free graph grammars [4]. In this paper, we give a graph grammar for two-dimensional grid graphs. The graph grammars for grid graphs is context-sensitive.

Keywords: context-sensitive graph grammar, grid graphs.

1 Introduction

Our interest is in a graph grammar for two-dimensional grid graphs. Graph grammar has succeeded in characterizing graph structures and generating several types of graphs [3, 8, 9, 10]. In [5] and [6], there are graph grammars that can derive ladders. Vigna and Ghezzi showed that a language of grid graphs could not be constructed using their context-free graph grammars [4]. In [1], Pavlidis also described that the set of grid graphs could not be generated using their context-free graph grammars.

On the other hand, Burosch and Laborde characterized grid graphs using the products of paths and showed that two-dimensional grids can be recognized in linear time by applying a combinatorial algorithm in [7].

Graph grammar characterization for grid graphs possibly derives characterization of grid graphs with multiple labels, such as business document tables [10] or symbol tables in program documents.

In this paper, We show that there is a context-sensitive graph grammar for two-dimensional grid graphs. This grammar derives grid graphs by synchronizing using state propagation among nodes in graph rewriting rules. Label propagation is based on state propagation of cellular automata (see e.g., [2]).

2 Notations and Definitions

First, we begin by reviewing some graph notations. For this paper, we consider directed graphs with node and edge labels. We also consider undirected graphs by extending the directed graphs.

Definition 1 [9] Let $\Sigma$ be an alphabet of node labels and $\Gamma$ an alphabet of edge labels. A graph over $\Sigma$ and $\Gamma$ is $G = (V, E, \psi)$, where $V$ is the finite set of nodes, $E \subseteq \{(v, l, w) | v, w \in V, l \in \Gamma\}$ is the set of edges, and $\psi : V \rightarrow \Sigma$ is the labeling function.

In this paper, we always assume that $G$ is connected. Graph $G = (V, E, \psi)$ is an undirected graph if, for every $(v, l, w) \in E$, there is also $(w, l, v) \in E$ [9].

**Definition 2** [7] A (undirected) graph is called a (two-dimensional) grid graph iff the graph consists of the product of two path graphs [7]. Let $P_{n1}$ and $P_{n2}$ be path graphs, then $P_{n1} \square P_{n2}$ denotes the product of $P_{n1}$ and $P_{n2}$. If $P_{n1}$ is a path graph with $n1$ nodes and $P_{n2}$ is a path graph with $n2$ nodes, then $T_{n1,n2}$ denotes a grid graph for the product of the $P_{n1}$ and $P_{n2}$.

Figure 1 shows a grid graph $T_{3,4}$.

![Figure 1: A grid graph.](image)

We now consider that two graphs are “isomorphic.” Two graphs $G_1 = (V_1, E_1, \psi_1)$ and $G_2 = (V_2, E_2, \psi_2)$ are isomorphic if there is a bijection $f: V_1 \rightarrow V_2$ such that $E_2 = \{(f(v), l, f(w)) | (v, l, w) \in E_1\}$ and, for all $v \in V_1$, $\psi_2(f(v)) = \psi_1(v)$ [9].

## 3 Context-Sensitive Graph Grammars

Next we consider graph transformations. Graphs are transformed based on production rules. We deal with the following productions, in which one production replaces a subgraph in one graph with other graph.

In this section, we modify edNCE graph grammars [9] and introduce context-sensitive graph grammars. The left-hand side of the production in the edNCE graph grammar is defined using a single node label. But the left-hand side of the production in our graph grammar is defined using a connected graph.

**Definition 3** (cf. [9]) A production $p: (D, C)$ is of the form $M \rightarrow (D, C)$, where $M$ is a graph over $\Sigma$ and $\Gamma$, $D$ is a graph over $\Sigma$ and $\Gamma$ called an embedded graph, and $C \subseteq \Sigma \times \Gamma \times \{\text{in, out}\}$, called a connection relation, is a set of connection instructions.

A production $p: M \rightarrow (D, C)$ removes a subgraph $M'$ of its applied graph $G$, such that $M$ and $M'$ are isomorphic, and replaces $M'$ with $D'$, such that $D'$ and $D$ are isomorphic and the set of nodes in $G$ and $D'$ are pairwise disjoint. Then the connection instructions of $C$ replace neighboring edges around $M'$. A connection instruction $(\alpha, \beta, x, \gamma, y, \text{in})$ of $C$ means that if there was an edge labeled $\beta$ from the node $x$ in $G$ for which $(D, C)$ is substituted to a node $w$ with label $\alpha$, then the embedding process will establish an edge labeled $\gamma$ from $y$ to $w$. Similarly, a connection instruction $(\alpha, \beta, x, \gamma, y, \text{out})$ of $C$ means that if there was an edge labeled $\alpha$ from a node $w$ with label $\alpha$ to the $x$ in $G$ for which $(D, C)$ is substituted, then the embedding process will establish an edge labeled $\gamma$ from $y$ to $w$.

That is, let $G = (V_G, E_G, \psi_G)$ and $H = (V_H, E_H, \psi_H)$ be graphs on $\Sigma$ and $\Gamma$. Let $p: M \rightarrow (D, C)$ be a production such that $V_G$ and $V_D$ are pairwise disjoint. Then $G \Rightarrow_p H$ holds for $p$ if there is a subgraph $M'$ of $G$ such that $M'$ and $M$ are isomorphic. That is we obtain $H$ from $G$ as follows: $V_H = (V_G - V_M') \cup V_D$, $E_H = \{(x, l, y) | x, y \in V_G - V_M' \} \cup E_D \cup \{(w, l, x) | \exists m \in \Gamma, \exists v \in V_M': (w, m, v) \in E_G, (\psi_G(w), m, v, l, x, in) \in C\} \cup \{(x, l, w) | \exists m \in \Gamma, \exists v \in V_M': (v, m, w) \in E_G, (\psi_G(w), m, v, l, x, out) \in C\}$, $\psi_H(x) = \psi_G(x)$ if $x \in (V_G - V_M')$, and $\psi_H(x) = \psi_D(x)$ if $x \in V_D$ (cf. [9]).

**Definition 4** [9]) A graph grammar is $GG = (\Sigma, \Delta, \Gamma, \Omega, P, S)$, where $\Sigma$ is the finite set of node labels, $\Delta \subseteq \Sigma$ is the set of terminal node labels, $\Gamma$ is the finite set of edge labels, $\Omega \subseteq \Gamma$ is the set of final edge labels, $P$ is the set of productions, and $S$ is the start graph.

We call context-sensitive graph grammar, in which the left-hand side of a production is a graph.

For productions $M_1 \rightarrow (D_1, \Gamma_1)$ and $M_2 \rightarrow (D_2, \Gamma_2)$, Two connection relations $\Gamma_1$ and $\Gamma_2$ are called isomorphic if there are isomorphism $f$ from the node set of $M_1$ to the node set of $M_2$ and $f'$ from the node set of $D_1$ to the node set of $D_2$ such that $C_2 = \{(\sigma, \beta, f(v), \gamma, f'(x), d) | (\sigma, \beta, v, \gamma, x, d) \in C_1\}$. 

Two productions $M_1 \rightarrow (D_1, C_1)$ and $M_2 \rightarrow (D_2, C_2)$ are called isomorphic if $M_1$ and $M_2$ are isomorphic graphs, $D_1$ and $D_2$ are isomorphic graphs, and $C_1$ and $C_2$ are isomorphic.

Let $GG = \{\Sigma, \Delta, \Gamma, \Omega, P, S\}$ be a graph grammar. Let $G$ and $H$ be graphs on $\Sigma$ and $\Gamma$ and let $p: M \rightarrow (D, C)$ be a production in $P$. Then $G \Rightarrow_p H$ is called a derivation step, and a sequence of derivation steps is called a derivation (cf. [9]).

We assume that $P$ does not contain distinct isomorphic productions in this grammar.

Figure 2 shows an example of a production and a derivation of a graph. Node labels $A, B, C, D$ are non-terminal, and label $\#$ is terminal. Every edge in the graphs shown in Figure 2 is unlabeled. Furthermore, a box is a nonterminal node, a black circle is a terminal node, an arrow is a directed edge, and a dotted arrow is part of a connection instruction. We use edge label $\#$ for unlabeled edges. Production $p$ is $M \rightarrow (D, C)$, where $M = (V, E, \psi)$ such that $V = \{x_1, x_2\}$, $E = \{(x_1, \#, x_2)\}$, $\psi(x_1) = A$, and $\psi(x_2) = B$, and $D = (V_D, E_D, \psi_D)$ such that $V_D = \{x_3, x_4, x_5\}$, $E_D = \{(x_3, \#, x_4), (x_4, \#, x_5), (x_5, \#, x_3)\}$, $\psi_D(x_3) = \#, \psi_D(x_4) = \#, \psi_D(x_5) = D$, and $C = \{c_1, c_2\}$ such that $c_1 = (\#, \#, x_1, \#, x_5, in)$ and $c_2 = (C, \#, x_2, \#, x_4, out)$.

Graph $H$ in Figure 2 is generated by applying $p$ to graph $G$.

**Definition 5** The language of graph grammar $GG = (\Sigma, \Delta, \Gamma, \Omega, P, S)$ is $L(GG) = \{g | g$ is derived from $S$ by $GG$, all node labels of $g$ are terminal node labels, and all edge labels of $g$ are final edge labels.$\}$

## 4 A Grid Graph Grammar

In this section, we construct a graph grammar for two-dimensional grid graphs and describe some graph grammar characteristics.

**Definition 6** A grid graph grammar $GG_G$ is defined as follows: $GG_G = (\Sigma_G, \Delta_G, \Gamma_G, \Omega_G, P_G, S_G)$, where $\Sigma_G = \{S, H, L, A, B, R, C, X, D, R', E, P_h, END, \#\}$, $\Delta_G = \{\#\}$, $\Gamma_G = \{\#\}$, $\Omega_G = \{\#\}$, $P_G$ is defined in Figure 3, and $S_G = (\{v\}, \phi, \psi_{S_G})$ such that $\psi_{S_G}(v) = S$.

Here we explain the labels in $P_G$. Label $S$ is the start node label. Label $L$ is the leftmost node label of grid graphs. Label $H$ is the label for making grid graphs that are not path graphs. Labels $R$ and $R'$ are the rightmost node labels of grid graphs. Labels $A, B, P_h$ are for growing grid graphs to the left. Labels $C$ and $X$ are for turning right. Label $END$ is the last nonterminal node label, and label $\#$ is the terminal node label.

Note that the unlabeled nodes and edges are labeled $\#$.

We show an example of a derivation of $GG_G$ by Figures 4, and 5 in the Appendix. This derivation is for the grid graph $T_{3,4}$. Figure 3 illustrates the productions of $GG_G$. In Figure 3, we denote the connection instructions of the productions using dotted lines.

**Theorem 1.** Graph $T$ is a grid graph iff $T$ is an element of $L(GG_G)$. 
Figure 3: Productions of $GG_G$. 
We can extend this graph grammar to graph grammars with multiple node labels.

Burosch and Laborde proposed a decision algorithm for two-dimensional grid graphs that runs in linear time for the number of nodes [7].

There are several applications of labeled grid graphs (e.g. symbol tables, bitmap image data etc.) that naturally relate labeled grid graphs. For example, the colors of pixels in bitmap images are represented by node labels.

Our graph grammar $GG_\sigma$ can be extended to several applications.

5 Conclusion

We constructed a graph grammar for two-dimensional grid graphs.

In the future, we will apply this grammar to grid graphs with multiple node labels such as financial statements. The results could possibly be applied to the parsing of grid graphs with multiple labels such as business document tables (e.g. [10, 11]) or symbol tables in program documents.

Acknowledgments

The authors would like to thank Professor Kimio Sugita of Tokai University and Associate Professor Koichi Yamazaki of Gunma University for their valuable discussions and helpful comments.

References

Appendix A: A Derivation Based on $GG_G$

This part shows the derivation process of the grid graph with three rows and four columns.

<table>
<thead>
<tr>
<th>An Example of a Derivation for Grid Graph $T_{3,4}$</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Derivation Diagram" /></td>
</tr>
</tbody>
</table>

Figure 4: A derivation based on $GG_G$ (1).
Figure 5: A derivation based on $GG_2 (2)$. 

An Example of a Derivation for Grid Graph $T_{3,4}$

$\Rightarrow P_{12}$

$\Rightarrow P_{13}$

$\Rightarrow P_{14}$

$\Rightarrow P_{15}$