社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2006－MPS－58（11）
2006／3／16

属性グラフ文法によるビジネス文書の定式化

志水　幸[†], 赤木　剛朗[†], 土田　賢省[††], 夜久　竹夫[†]

ビジネス文書の定式化は, 電子商取引や電子政府などの進展に伴って重要な課題になっている. 本論文では, 代表的なビジネス文書の 1 つである財務諸表に対して, グラフ文法により計算範囲の柔軟な指定も可能とする定式化を行うことを目的とする. 財務諸表の定式化のためには, 項目の順序や項目の数を規定する事と, 小計などの計算方法を規定する事が必要である. そのため, 財務諸表の項目の順序や数の規定を可能とする文脈依存グラフ文法を構築し, そのグラフ文法に計算方法を規定するための属性を付け加える.

A Formalization of Business Documents by an Attribute Graph Grammar

Miyuki Shimizu[†], Goro Akagi[†], Kensei Tsuchida[††], and Takeo Yaku[†]

The formalization of business documents has become an important subject with the progress of e-commerce and e-government. In this paper, we formalize the structure of financial statement, which is a sort of business document, by using graph grammar. To formalize financial statements, we specify the order and the number of items that appear and calculation methods. Therefore, we propose a context-sensitive graph grammar which enables the specification of the order and the number of items to formalize financial statements. We also append attributes to the context-sensitive graph grammar to specify calculation methods.

1 Introduction

The formalization of business documents has become an important subject with the progress of e-commerce and e-government. Recently, business documents have been formalized by XBRL (eXtensible Business Reporting Language) [8]. However, in the past formalizations including XBRL, specifying the range of calculations such as subtotal and average by course of columns and rows are difficult. Moreover, table calculation is implemented in a fourth generarion languages, e.g., Excel, however, verification of the order and the number of items (cf. [3]) and a flexible specifications of the range of the calculations are disable.

To cope with these problems, we use a Context-Sensitive Graph Gammar (CSGG for short). More precisely, production rules in the CSGG enable verification by locally changing the graph representation of tables, and moreover, we can also use the function of automatic table culculation by adding semantic rules specifying culculation methods for attribute-values to production rules.

In this paper, we formalize the structure of financial statement, which is a sort of business document, by using CSGG, and our formalization enables us to specify the ranges of calculations with no restriction (cf. [5]). To do so, we first propose a CSGG which provides the order and the number of items in homogeneous parts of financial statements. Secondly, we define an attribute graph grammar by adding semantic rules to the production rules of the CSGG.

2 Preliminary : Graph Representation of Financial Statements

Applications of CSGG have been developed by several authors, e.g., [6], [7], [9]. In particular, Arita et al [9] construct a CSGG characterising and generating two-demensional grid graphs. we modify the definition of the CSGG in [9] to formalize financial statement in Sect. 3. To formulate CSGG, we employ the notion of marked graph whose definition is as follows.

Definition 2.1 A *marked graph* is a system (K, R, k, r), where K is a finite set of nodes, $K \neq \emptyset$, $R \subset K \times K$, $k : K \to V$ a mapping for marking the nodes, $r : R \to M$ a mapping for labeling the edges. □

Moreover, we also use the equivalence of marked graph defined by

Definition 2.2 Two marked graphs $G_i = (K_i, R_i, k_i, r_i)$, $i \in \{1, 2\}$ are *equal* iff there exists a bijection $f : K_1 \to K_2$ such that f yields a bijection of R_1 to R_2, every $a \in K_1$ has the same label as $f(a) \in K_2$ and vice versa, and every $(u, v) \in R_1$ has the same label as $(f(u), f(v))$ and vice versa. □

Now, as in [6], we formalize graphs representing the homogeneous parts in financial statements such

*The results are partially appeared in "Miyuki Shimizu, Kinsei Tsuchida, and Takeo Yaku, A Formalization of Buseness Documents by Graph Grammars (in japanese), ISPJ SIG Technical Report, 2004-IS-88, 10, pp.66-73, 2004."

[†]日本大学文理学部
College of Humanities and Sciences, Nihon University
[††]東洋大学工学部
School of Engineering, Toyo University

as Fig. 1. To do so, the homogeneous parts will be first expressed as graphs. Fig. 2 and Fig. 3 are a typical example of a financial statement and its graph representation, respectively. Each node in Fig. 3 coresponds to each cell of the financial statement in Fig. 2, and marks of nodes stand for items in the financial statement. Marks with prime, e.g., DO', are given to items for numerical data and marks with double prime, e.g., DO'', are given to items for the results of culculations. Moreover, the label lf of an edge (a, b) means that the two cells coresponding to the nodes a and b are horizontally adjacent to each other, and the edge label ov similarly means the vertical adjacency of the two cells.

	2002	2003		change
Domestic Operations				
Overseas Operations				
Sub total			⋮	
Domestic Operations				
Overseas Operations				
Sub total				
...				

Figure 1: a homogeneous part of the financial statement

	2002	2003	change
Domestic Operations			
Overseas Operations			
Sub total			
Domestic Operations			
Overseas Operations			
Sub total			

Figure 2: a part of financial statement

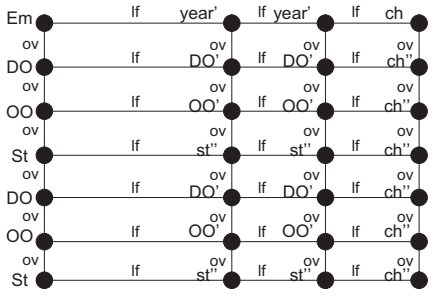


Figure 3: a graph expression of Fig.2

3 Graph Grammar of Homogeneous Part of Financial Statements

Formalizing financial statements in context free graph grammars [1] that change one node is difficult because relationships among items in financial statements are complicated. Moreover, the frames of homogeneous parts in financial statements can be expressed by two-dimensional grid graphs, but it is proved that there exists a grid graph G such that G cannot be generated by using NLC grammars, which is a sort of context-free graph grammar (see [2]). Therefore, we employ the notion of CSGG to formalize financial statements. The production rules transform a marked graph (not a node) into a marked graph. We define a CSGG by modifying the definition of the CSGG [9] as follows.

Definition 3.1 A *context-sensitive production* is a tuple $p = (A, H, p^c)$, where $A = (K_a, R_a, k_a, r_a)$ and $H = (K_h, R_h, k_h, r_h)$ are nonempty connected marked graphs (cf. [7]), $p^c : (M, N \cup T) \to K_h$, M is the set of all labels for edges, and N and T denote the set of nonterminal symbols, respectively, such that $|K_a| \leqq |K_h|$. □

Definition 3.2 For marked graphs $G = (K, R, k, r)$ and $G' = (K', R', k', r')$ the relationship $G \underset{p}{\Longrightarrow} G'$ holds for $p = (A, H, p^c)$ with $H = (K_h, H_h, k_h, r_h)$ iff G is blown up to G' by replacing graph A in G by H and replacing the edges connected to A according to p^c, i.e.,

(i) there exists a graph with $A = (K_a, R_a, k_a, r_a)$,

(ii) $K' = (K - K_a) \cup K_h$,

(iii) $k'(a) = k(a)$ if $a \in (K - K_a)$; $k'(a) = k_h(a)$ if $a \in K_h$,

(iv) $R' = R - \{(a, i) \mid i \in K_a\} - \{(i, j) \mid i, j \in K_a\} \cup R_h \cup \{(a, x) \mid i \in K_a, \bigvee_{(a,i)\in R} k(i) = m, r((a, i)) = l \text{ and } p^c(l, m) = x \in K_h\}$,

(v) $r'((a, b)) = r((a, b))$ if $(a, b) \in R$; $r'((a, b)) = r_h((a, b))$ if $(a,b) \in R_h$, $m \in M$ for all edges between H and the rest of G which are inherited from an edge in G labeled by m. □

Definition 3.3 A *CSGG* is a system $GG_F = (N_F, T_F, M_F, P_F, S_F)$, where N_F is the set of nonterminal symbols, T_F is the set of terminal symbols, M_F is a finite set of labels for the edges, P_F is a finite set of context sensitive productions $p = (A, H, p^c)$, and $S_F \in N_F$ is the startsymbol, i.e., the startgraph for GG_F. □

We denote by $\overset{*}{\Rightarrow}$ the reflexive and transitive closure of the symbol $\Rightarrow$. Graph G' is said to be derived by the graph grammar GG if $G \underset{GG}{\overset{*}{\Longrightarrow}} G'$ (GG and $*$ can be omited). We define a *(context-sensitive) graph language* $L(GG) = \{G | S \underset{GG}{\overset{*}{\Longrightarrow}} G\}$ as in the string grammar.

Next, we define a graph grammar equivalent to a homogeneous part of financial statements as follows.

Grammar 3.4 A *financial statement CSGG* is a system $GG_{F1} = (N_{F1}, T_{F1}, M_{F1}, P_{F1}, S_{F1})$, where $N_{F1} = \{S, A, B, C, D, E, F, G, H, I, J, K, L, N, P, Q, R\}$, $T_{F1} = \{ DO, OO, St, DO', OO', St', year', ch, ch', Em \}$, $M_{F1} = \{①, ..., ⑬, lf, ov\}$, and $S_{F1} = S$. P_{F1} is shown in Appendix 1. □

The terminal label DO stands for "Domestic Operations", OO for "Overseas Operations", St for "Sub total", ch for "change", and Em for an empty cell .

In Fig. 4, we show a couple of the production rules. Each production rules replaces a marked graph on the left-hand side of the arrow by a marked graph on the right-hand side of the arrow. Each dashed line denotes an edge (a, b) between a node $a \in K_a$ (resp., $a \in K_h$) and a node $b \notin K_a$ (resp., $b \notin K_h$) adjacent to K_a (resp., K_h). For example, the production rule p2 makes the edges labeled ①, ②, ③, ④, which linked to the node marked with A, link to the node marked with B.

Furthermore, labels including such, e.g., label1/label2 mean that the edge is first labeled label1, and the label is replaced by label2 after applying a production rule. Applications of production rules can be shown as in Fig. 5.

Here we give two remarks on the correspondence from $L(GG_F)$ into financial statements.

Remark 1 If T is derived by the grammar GG_{F1}, then T is a financial statement as shown in Fig. 2.

Remark 2 If T is a financial statement as shown in Fig. 2, then derivation of T by GG_{F1} is unique . Indeed, p15 and p16 transmit symbol D from right to left, and the p36 and p37 also do so as in the case of p15 and p16.

The detail of proofs for the claims in the two remarks above is left to the forthcomming papers.

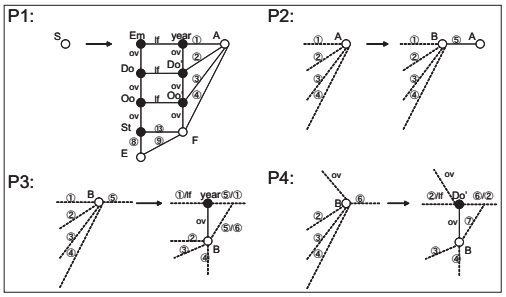


Figure 4: A part of productions in GG_F

4 Attribute Graph Grammar[4]

In this section, we define semantic rules and corrspond them to the production rules for the CSGG GG_{F1}.

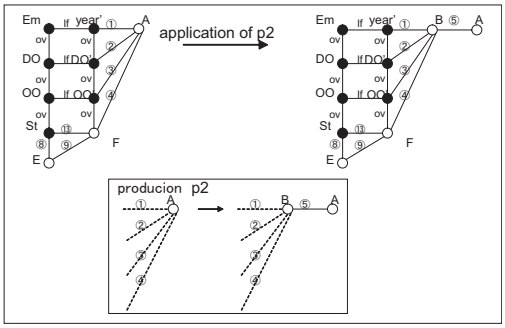


Figure 5: Example of application of productions in GG_F

4.1 Attribute Graph Grammar for Financial Statements

Definition4.1 A context-sensitive *attribute graph grammar* is a tuple $AGG_{F1} = \langle GG_{F1}, \mathcal{A}, F \rangle$, where

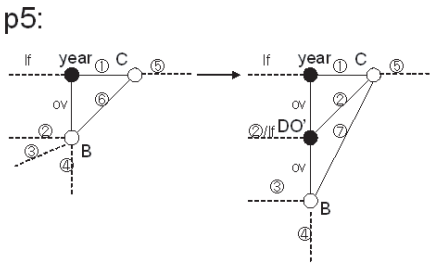
- $GG_{F1} = (N_{F1}, T_{F1}, M_{F1}, P_{F1}, S_{F1})$ defined in grammar 3.3 is called basic CSGG of AGG_{F1}.
- For $x \in N_{F1} \cup T_{F1}$, let $\mathcal{I}(x)$ be the set of inherit attributes and let $\mathcal{S}(x)$ be the set of synthesise attributes. Let $\mathcal{I} = \bigcup_{x \in V_{F1} \cup T_{F1}} \mathcal{I}(x)$ and $\mathcal{S} = \bigcup_{x \in V_{F1} \cup T_{F1}} \mathcal{S}(x)$. Let $\mathcal{A}(x) = \mathcal{I}(x) \cup \mathcal{S}(x)$. Let $\mathcal{A} = \bigcup_{x \in V_{F1} \cup T_{F1}} \mathcal{A}(x)$ be an attribute set of AGG_{F1}. Moreover, $a(x)$ stands for an attribute a of x and $\mathcal{V}(a)$ stands for a set of values that a holds.
- Each production rule $p_i = (A_i, H_i, p_i^c)$ in P_{F1} has semantic rule sequence s_i. s_i defines attribute of $\bigcup_{x \in V(A_i)} \mathcal{S}(x) \cup \bigcup_{x \in V(H_i)} \mathcal{I}(x)$. Set $F = \bigcup s_i$ are called semantic rule set.
- $\mathcal{I} = \{row, col, basisSt, basisDO\}$ $basisOO, sum, value, valueDO\}$, $\mathcal{S} = \{\}$ □

4.2 Application of Semantic Rules

Semantic rules are the sets of the functions of attribute values. Moreover, we add the following functions to our attribute graph grammar to access input data of tables.

Function

- `get_value(c)` returns the value which inputted in the cell c.
- `inherit(a, b)` hands over all values of inherit attribute of the cell a have to the cell b.



$inherit(B_1, DO')$
$value(DO') = get\_value(DO')$
$basisDO(DO') = basisDO(B_1)$
$sum(DO') = value(DO')$
$row(B_2) = row(B_1) + 1, col(B_2) = row(B_1),$
$sum(B_2) = value(B_2) + sum(DO')$

Figure 6: Production rule p_5 and its semantic rules

4.3 Semantic Rules of AGG_{F1} and its Application

We show a production rule p_5 and the corresponding semantic rule in Fig. 6, where $attr(node)$ denotes the value of the attribute $''attr''$ of the node marked with $node$. In case there are the "samelabel" in both side, we write the label of left side by "samelabel1", and right by "samelabel2".

4.4 Application of an Attribute Graph Grammar

We consider that the way to apply this attribute graph grammar at generation support of financial statements.

4.4.1 Generation of Table Format and Automatic Table Calculation

The grammars are equivalent to the generarion system of table format. For example, when "Term" "Number of Department", and other values are given as parameter to GG_{F1} it generates a table format in which each cell has appropriate attributes, for the parameters. When a user fills values in the cells, values of "subtotal" and "rate of increase" are calculated automatically and written in right cells.

5 Conclusion

In this paper, we converted a homogeneous part of a financial statement to graph, and constructed a graph grammar which produces its graph. This grammar is CSGG and have 38 productions. Moreover, we append attributes on grammar, so we can produce financial statements that have each attribute in each cell. Result, we can add new functions as table calculation and verification on graph grammar that produced the framework of table only.

As future works, we will convert financial statement include a heterogeneous part to graph and construct a CSGG which produces its graph. Moreover, For a graph grammar include a heterogeneous part of a financial statement, we will append attributes.

References

[1] Reinhold Franck, A Class of Linerly Parsable Graph Grammars, Acta Informatica, 10, pp.175–201, 1978.

[2] D.Janssens, G. Rozanberg, On the Structure of Node-Label-Controlled Graph Languages, INFORMATION SCIENCES 20, pp.191–216, 1980.

[3] Amihood Amir, Gary Benson, Martin Farach, An Alphabet Independent Approach to Two Dimensional Matching, SIAM J. Comp, 1994.

[4] Takashi Imaizumi, Takuya Katayama, Masataka sakki, Youiti Shinoda, Ikuo Nakata, Teturou nishino, Hiroyuki Matsuda, An Introduction To Attribute Grammar (in japanese), SAIENSU-SHA, Co.,Ltd, 1996.

[5] Margaret Burnett, Andrei Sheretov, Gregg Rothermel, Scaling Up a "What You See Is What You Test" Methodology to Spredsheet Grids, IEEE Symp. on Visual Languages, pp.30-37, 1999.

[6] Tomokazu ARITA, Kiyonobu TOMIYAMA, Takeo YAKU, Youzou MIYADERA, Kimio SUGITA, Kensei TSUCHIDA, Syntactic Processing of Diagrams by Graph Grammars, Proc. 16th IFIP World Computer Congress : Internat. Conf. Software (ICS2000), pp.145-151, 2000.

[7] Yoshihiro ADACHI, Suguru KOBAYASHI, Kensei TSUCHIDA, Takeo YAKU, Block Diagram Grammar (in japanese), IEICE Transaction, Vol.J83-D-I No.1, pp.45-54, 2000.

[8] Japanese Industrial Standards Committee(JISC), eXtensible Business Reporting Language 2.0 Specification (in japanese), TR X 0084, p.46, Japanese standards association, 2003.

[9] Tomokazu ARITA, Kensei TSUCHIDA, Takeo YAKU, Syntactic Characterization of Two-Dimensional Grid Graphs, IEICE TRANS. VolE82, to appear.