表形式のプログラム仕様書のためのグラフ文法

A Graph Grammar for Tabular Program Specification Forms

有田 友和[†] 杉田 公生[††] 土田 賢省[†††] 夜久 竹夫[†]

Tomokazu Arita Kimio Sugita Kensei Tsuchida Takeo Yaku

日本大学[†] 東海大学[††] 東洋大学[†††]

Nihon University Tokai University Toyo University

1 Introduction

Graph grammars have been studied for *program diagrams*[1, 2]. This paper deals with *tabular forms* for program specification and its syntactic definition with respect to mechanical drawing.

Items concerning program specification documents were discussed in a general sense in [3]. You can refer to [4] for the program specification documents in tabular forms. The order of items and visual structure of items concerning program specifications were partly introduced in [6] by precedence graph grammars [1].

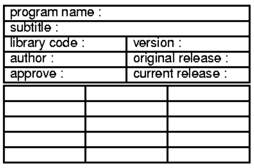
The purpose of this paper is to introduce formal definition of whole program specification forms by attribute NCE graph grammars with respect to syntactic manipulation and mechanical drawing.

2 Tabular Forms and Marked Graphs

We represent specification forms by means of *graphs.*

We use a *nested diagram* for the specification form and a *marked graph* for the nested diagram. The *mark* of a *node* of the marked graph represents an item of the specification form. The *label* of an *edge* of the marked graph is given a positional relationship between items.

Fig.1 shows part of a specification form. The upper part is a *header* portion of the form which has the structure of a marked *tree.* The lower part is a body portion of the form such as symbol tables having a tabular form structure, and is represented by a marked *tessellation* graph. Fig.2 shows a *nested* diagram for a specification form shown in Fig.1. Fig.3 shows a marked graph for the nested diagram shown in Fig.2.



program name :		
subtitle :		
library code :	version :	
author :	original release :	
approve :	current release :	

Fig.1 A tabular specification form

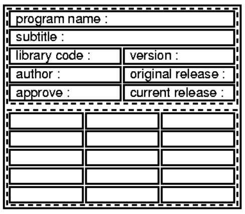


Fig. 2 A nested diagram for the tabular specification form in Fig. 1.

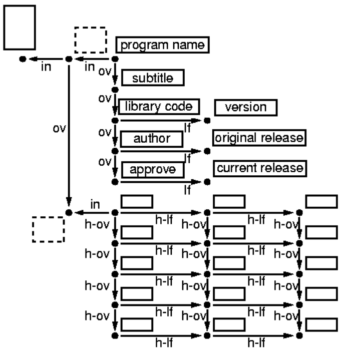


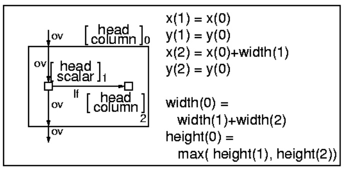
Fig.3 A marked graph for the nested diagram in Fig. 2.

3 Attribute Graph Grammars for Tabular Forms

We use a context sensitive NCE graph grammar as a model ($cf.$[5]).

We construct an *attribute graph grammar* G_1 for the tree structural part, and its example is shown in Fig.4. Moreover, we construct an attribute graph grammar G_2 for the tessellation part, and its example is shown in Fig.5. We note that the tessellation part can not be represented by precedence graph grammars [1]. The total sizes of G_1 and G_2 are shown in Table1.

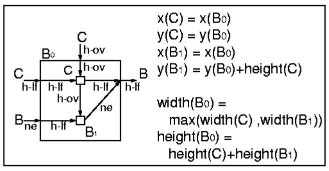
Claim The precedence relation in G_1 is similarly defined as in [1, 6]. The precedence table of G_1 has 15,330 cells in the sense of [1]. □



x(1) = x(0)
y(1) = y(0)
x(2) = x(0)+width(1)
y(2) = y(0)

width(0) =
width(1)+width(2)
height(0) =
max(height(1), height(2))

Fig. 4 A *production* with attribute rules of G_1



$x(C) = x(B_0)$
$y(C) = y(B_0)$
$x(B_1) = x(B_0)$
$y(B_1) = y(B_0)+height(C)$

$width(B_0) =$
$max(width(C) ,width(B_1))$
$height(B_0) =$
$height(C)+height(B_1)$

Fig. 5 A production with attribute rules of G_2

Numbers	G_1	G_2
Productions	280	34
Attribute rules	1218	232

Table 1 The sizes of G_1 and G_2

4 Conclusion

We suggested a fomalization of program specificiation forms by means of a modified NCE graph grammar for both the logical strucutre and visual structure. We are now developing a software document supporting system utilizing our suggested approach in this paper. We thanks very much Mr. K.Tomiyama of Nihon University and Mr. S.Kanai for valuable suggestions.

Reference

[1]Reinhold Franck, A Class of Linearly Parsable Graph Grammars, *Acta Infomatica 10*, 175-201 , 1978.

[2]G. Engels, R. Call, M. Nagl, *et al.*, Software specification using graph grammars, *Computing 31*, 317-346 ,(1983).

[3]ISO6592-1985, Guidelines for the documentation of computer-based application systems, 1985.

[4]K. Sugita, Y. Adachi, Y. Miyadera, K. Tsuchida and T. Yaku, *Proc. of Advanced Software Mechanisms for Computer-Aided Instruction Information Literacy APEC-CIL'97*, 1997.

[5]Grsegorz Rozenberg, Handbook of Graph Grammar and Computing by Graph Transformation,World Scientific, 1-94, 1996.

[6]T.Arita, Y.Adachi, K.Sugita, K.Tsuchida, T.Yaku, Formalization of Software Documents Using a Precedence Graph Grammar, Proc. 1999 Inform. Systems Soc. Conf. IEICE D-3-2, p22,1999.