Attribute Graph Grammars and Tabular Forms

by
Tomokazu ARITA

A Thesis Submitted to
Earth Information Mathematical Sciences,
Graduate School of Integrated Basic Sciences,
Nihon University
in Fulfillment of the Requirements
of the Degree of Master of Science

March, 2000

Preface

This thesis characterizes graph grammars which provide formal definition of program documentation tabular forms with respect to syntactic manipulation and mechanical drawing. We propose an attribute context-free graph grammar with 280 rewriting rules and 1248 attribute rules for ISO 6592 based nested program forms with 137 items. The grammar is shown to have precedence property [1] by 5376 relations over the marks. Furthermore, we consider context-sensitive graph grammars for tessellation tabular forms.

Keywords graph grammars, program documents, form layout.

Acknowledgments

I would like to express my greatest gratitudes for Professor Takeo Yaku at Nihon University, who has been encouraging and advising me throughout my whole study to complete this thesis.

I would also like to express thanks to Professor Yoshihiro Adachi at Toyo University, Professor Kimio Sugita at Tokai University, Professor Kensei Tsuchida at Toyo University, and Professor Youzou Miyadera at Tokyo Gakugei University for their encouragements and valuable suggestions.

I would also like to thank Mr. Akira Matsumoto at NTT, Mr. Kiyonobu Tomiyama and Mr. Kenji Ruise at Nihon Univerity, for I could have fruitful discussions with them.

I would also like to thank Mr. Shinichiro Kanai for his advice.

Tomokazu ARITA

Earth Information Mathematical Sciences,
Graduate School of
Integrated Baic Sciences,
Nihon University
3-25-40 Sakura-jyosui, Setagaya-ku,
Tokyo 156-8550, JAPAN

Contents

Chapter 1

Introduction

Graph grammars have been studied and utilized, by several authors, for their possible association with program diagrams, generation of general diagrams and computer aided design for the industrial objects. (see e.g. [1],[2])

This paper deals with tabular forms for program specification documents and its syntactic definition with respect to the mechanical drawing. Items in program specification documents were generally listed in [3]. The program specification documents are usually represented by tabular forms [5].

We came to notice that tabular forms can generally be represented by graphs. Thus, in this paper we regard the tabular forms as nested diagrams and represent nested diagrams by marked graphs.

In [1], Franck used marked graphs for nested diagrams, introduced a precedence graph grammar for the marked graphs and formalized parsing of nested diagrams. Nishino [4] introduced an attribute graph grammar with respect to a drawing problem of tree-like diagrams and formalized transformation of tree-like diagrams. In [4], the drawing problems were specified by semantic rules of attributes. We have also studied syntactic and algorithmic manipulation of diagrams [6], [7] [8]. The purpose of this paper is to characterize graph grammars which provide formal definition of program specification forms with respect to syntactic manipulation and mechanical drawing.

This thesis is organized as follows:
In Chapter 2, we introduce a program documentation system Hiform96 [5] and review definitions of a context-free graph grammar and precedence grammar [1].

In Chapter 3, we introduce definitions of an attribute graph grammar and an attribute precedence graph grammar. Then we show an attribute precedence graph grammar for Hiform, and illustrate how to solve a layout problem of Hiform by using attribute evaluatuions.

In Chapter 4, we review a definition of NCE graph grammar. Then we consider tessellation forms and show an attribute graph grammar for them.

In Chapter 5, we summarize our reults.

Chapter 2

Preliminaries

2.1 Program Documentation Language Hiform96

We introduce here a program documentation system called Hiform96 [6] based on ISO6592 [3].

The International Organization for Standardization issued a guideline in ISO6592 and described all items in program documentation in Annex A, B and C. We considered the ISO6592 items and introduced Hiform96, which includes all items defined in these Annexes. Hiform96 is defined by 17 types of forms.

Hiform96 was originally developed for the purpose of the programming education. Hiform document is a collection of tabular forms. Using tabular forms, one can understand at a glance what information should be obtained, what information is lacking, how a project is proceeding, and how to process and maintain the software. Besides these characteristics, the tabular form can include various description style such as letters and diagrams.

The following Fig 2.1 shows a Hiform96 program documentation form.

The order among tabular forms is defined by a context-free string grammar [5].

program name : hanoi_main	**A**
subtitle : hanoi	**General document**
library code : cs - 2000 - 01	**version :** 1.0
author : Tomokazu Arita	**original release :** 1999/12/22
approver :	**current release :** 2000/01/28
key words : Hanoi Tower	**CR-code :**
scope : Fundamental	
variant :	
language : Java	**software req. :** JDK 1.2
operation : **Interactive batch realtime**	**hardware req. :**
references :	

function : 1. list and explanation of input data or parameter,
2. list and explanation of output data or return value.

1. list and explanation of input data.

```
int  n;   [ Number of Plates ]
String  target;        [ Target Symbol ]
String  work;          [ Working Symbol ]
String  destination;   [ Destination Symbol ]
```

2. list and explanation of output data and return value.

```
output data  :  No.  to be moved: Source Symbol  ->  Destination Symbol
return value :  void
```

example :

1. Example of Operation

```
hanoi(5, A, B, C)
```

2. Example of Output

```
1:  A -> C
2:  A -> B
1:  C -> B
3:  A -> C
1:  B -> A
. . . . . .
```

Fig. 2.1 A program documentation of Hiform96.

2.2 Nested Diagrams for Tabular Forms

We use a nested diagram as a formalization of a tabular form document. We express a structure of a tabular form document by using a nested diagram. A structure is expressed as relations between items of a tabular form document, not expressed as arrangements of lines and characters. The following Fig 2.2 illustrates the nested diagram that represents a tabular form.

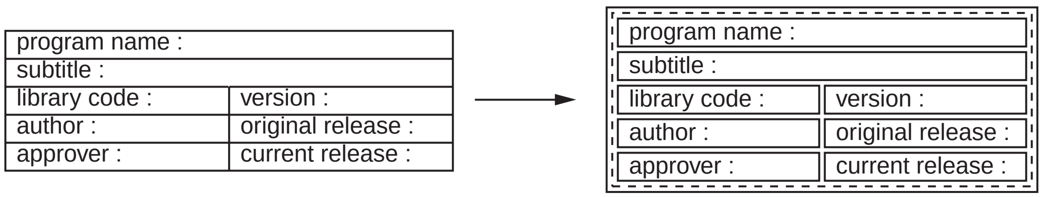


Fig. 2.2 A tabular form and its corresponding nested diagram.

2.3 Marked Graphs for Nested Diagrams

A marked graph is decided as follows: (1)A mark of marked graph shows an item of a nested diagram. (2)An edge label shows relations between items.

We introduce a marked graph for a nested diagram as an example. An edge label shows relations between items. An edge label ”lf” denotes ‘left of’, ”ov” denotes ‘over’, and ”in” denotes ‘within’.

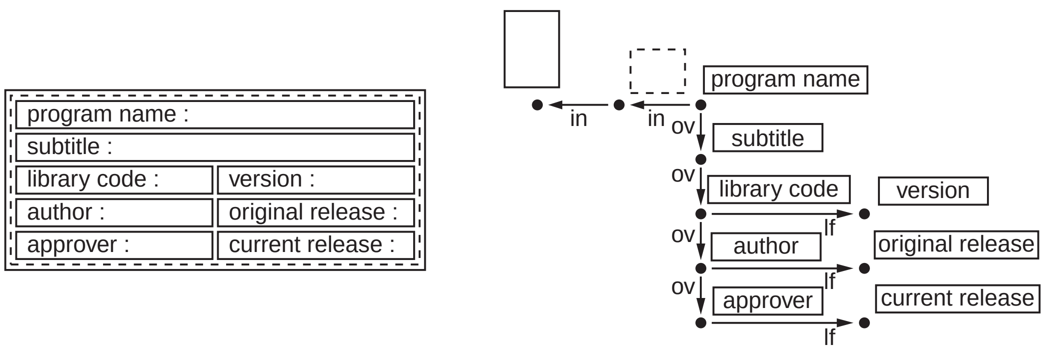


Fig. 2.3 A nested diagram shown in Fig 2.2 and its corresponding marked graph.

We introduce a definition of a marked graph [1].

Definition 2.3.1 [1]. *A marked graph* is a system (K, R, k, r) where K is a finite set of *nodes*, $K \neq \emptyset$, $R \subseteq K \times K$ is a finite set of *edges*, $k : K \to V$ a mapping for *marking* the nodes (V is a finite set of *alphabet*), $r : R \to M$ a mapping for *labeling* the edges (M is a finite set of all *labels* for edges).

2.4 Context-Free Graph Grammar

We survey here context-free graph grammars [1] and precedence grammars [1]

Definition 2.4.1 [1]. *A (context-free) production* is a 4-tuple $p = (A, H, p^e, p^s)$, where A is a single node graph (the left-hand side of p), $H = (K_h, R_h, k_h, r_h)$ is a nonempty graph (the right-hand side of p), and $p^e, p^s : M \to K_h$ are partial functions where M is the set of all labels for edges. □

The following Fig. 2.4 shows an example of a production.

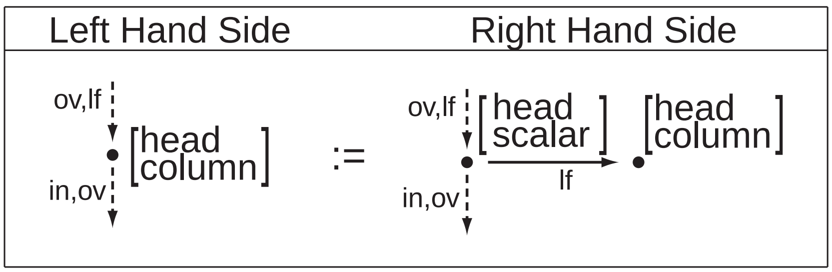


Fig. 2.4 An example of a production.

Definition 2.4.2 [1]. *A context-free graph grammar* is a system $GG = (V, T, M, P, S)$, where V is a finite set of alphabet, i.e. a set of symbols for labeling the nodes, $T \subset V$ is a set of the *terminal* symbols, M is a finite set of labels for the edges, P is a finite set of productions of the form $p = (A, H, p^e, p^s)$ explained above, $S \in V - T$ is the *start symbol*, i.e. the *start graph* for GG. □

2.5 Precedence Relation and Precedence Grammar

Notation 2.5.1 [1]. For $m \in M$ let

$$\doteq_{\mathbf{m}} \stackrel{def}{=} \left\{ (A,B) \,\middle|\, \begin{array}{l} A, B \in V \text{ and there exists a rule with an edge } (x,y) \\ \text{on the right-hand side where } x \text{ is labeled by } A, \\ y \text{ is labeled by } B \text{ and } (x,y) \text{ has label } m. \end{array} \right\}$$

$$\rightarrow_{\mathbf{m}} \stackrel{def}{=} \left\{ (A,B) \,\middle|\, \begin{array}{l} A, B \in V \text{ and there is a rule } p = (A, H, p^e, p^s) \\ \text{and } B \text{ is the label of the node } p^e(m) \text{ in } H \end{array} \right\}$$

$$\leftarrow_{\mathbf{m}} \stackrel{def}{=} \left\{ (B,A) \,\middle|\, \begin{array}{l} A, B \in V \text{ and there is a rule } p = (A, H, p^e, p^s) \\ \text{and } B \text{ is the label of the node } p^s(m) \text{ in } H \end{array} \right\}$$

□

Notation 2.5.2 [1]. For $m \in M$ let

$\lessdot_{\mathbf{m}} \stackrel{def}{=} \doteq_m \cdot \stackrel{+}{\rightarrow}_m$

$\gtrdot_{\mathbf{m}} \stackrel{def}{=} \stackrel{+}{\leftarrow}_m \cdot \doteq_m$

$<\!\cdot\!>_{\mathbf{m}} \stackrel{def}{=} \stackrel{+}{\leftarrow}_m \cdot \doteq_m \cdot \stackrel{+}{\rightarrow}_m$

where $+$ denotes transitive closure.

□

Precedence Relations are $conflictless$ if and only if for every $m \in M$ the relations $\lessdot_m, \doteq_m, \gtrdot_m$ and $<\!\cdot\!>_m$ are pairwise disjoint [1].

Definition 2.5.3 [1]. A context-free graph grammar is called a *precedence grammar* if and only if (i) the precedence relations are conflictless. (ii) all rules are uniquely invertible. (iii) there is no reflexive nonterminal symbol in the grammar.

□

Chapter 3

Attribute Precedence Graph Grammar for Hiform

3.1 Definitions for Atribute Graph Grammar

We introduce an another type of graph grammars for formalization of tabular forms based on [1] and [4].

Definition 3.1.1 (cf. [1], [4]) *An attribute graph grammar* is a 3-tuple $AGG = \langle GG, Att, F \rangle$, where

1. $GG = (V, T, M, P, S)$ is called an *underlying context-free graph grammar* of AGG. Each production p in P is denoted by $p = (A, H, p^e, p^s)$. $Lab(H)$ denotes the set of all occurrences of the node symbols labeling the nodes in the graph H.

2. Each node symbol $X \in V$ of GG has two disjoint finite sets $Inh(X)$ and $Syn(X)$ of *inherited* and *synthesized attributes*, respectively. We denote the set of all attributes of nonterminal node symbols X by $Att(X) = Inh(X) \cup Syn(X)$. $Att = \bigcup_{X \in V} Att(X)$ is called the set of attributes of AGG. We assume that $Inh(S) = \emptyset$. An attribute a of X is denoted by $a(X)$, and set of possible values of a is denoted by $V(a)$.

3. Associated with each production $p = (X_0, H, p^e, p^s) \in P$ is a set F_p of *semantic rules* which define all the attributes in $Syn(X_0) \cup \bigcup_{X \in Lab(H)} Inh(X)$. A

semantic rule defining an attribute $a_0(X_{i_0})$ has the form $a_0(X_{i_0}) := f(a_1(X_{i_1})$, ..., $a_m(X_{i_m}))$, $0 \leq i_j \leq |Lab(H)|$, $X_{i_j} \in Lab(H), 0 \leq j \leq m$. Here $|Lab(H)|$ denotes the cardinality of the set $Lab(H)$, and f is a mapping from $V(a_1(X_{i_1}) \times$ ... $\times a_m(Xi_m))$ into $V(a_0(X_{i_0}))$. In this situation, we say that $a_0(X_{i_0})$ depends on $a_j(X_{i_j})$ for $j, 1 \leq j \leq m$ in p. The set $F = \bigcup_{p \in P} F_p$ is called the *set of semantic rules* of AGG.

□

Definition 3.1.2 An attribute graph grammar $AGG = \langle GG, Att, F \rangle$ is *an attribute precedence graph grammar (APGG)* iff GG is a precedence graph grammar. □

3.2 An Attribute Precedence Graph Grammar for Hiform

We propose an attribute graph grammar which characterize the Hiform documents. The characterized forms are called Hiform2000.

The grammar which formalizes Hiform2000 is called Hiform Attribute Graph Grammar(HFAGG). We show productions of HFAGG in Appendix A. HFAGG consists of 280 productions. The mark of the start graph is ”[struct]”.

We also construct 1248 semantic rules of HFAGG as shown in Appendix A. Each production is associated with semantic rules. These semantic rules are mainly used for evaluating the positions and the sizes of items. In the definition of each production, a number added on the bottom right of the mark is the identification number in the productions [6].

Proposition 1 . The grammar HFAGG above is an attribute precedence graph grammar.

Proof. We can construct 5376 relations over the marks in HFAGG as shown in Appendix A. The relation are shown to be pairwise disjoint.

Remark. We can implement a linear time parser[1] for the underlying graph grammar of HFAGG.

3.3 Layout Problem of Hiform

Layout problems of nested diagrams are solved by attribute evaluations [4]. We use attributes which are *x*, *y*, *width* and *height*. Symbols *x* and *y* are used to calculate *x* coordinate and *y* coordinate, respectively. And *width* and *height* are also used to calculate width and height, respectively. We illustrate a process of an attribute evaluation in Fig. 3.1. The yield of the derivation tree of Fig.3.1 is a program document shown in Fig 3.2. Thus, we have:

Proposition 2 Attributes in HFAGG are evaluated in linear time. □

In Fg 3.1, we set values of items as follows: WIDTH_pname and WIDTH_stitle are 2 respectively. WIDTH_lcode, WIDTH_version, WIDTH_author, WIDTH_approver, WIDTH_orelease, and WIDTH_crelease are 1 respectively. The height of each item is 1. And all margins are 0.

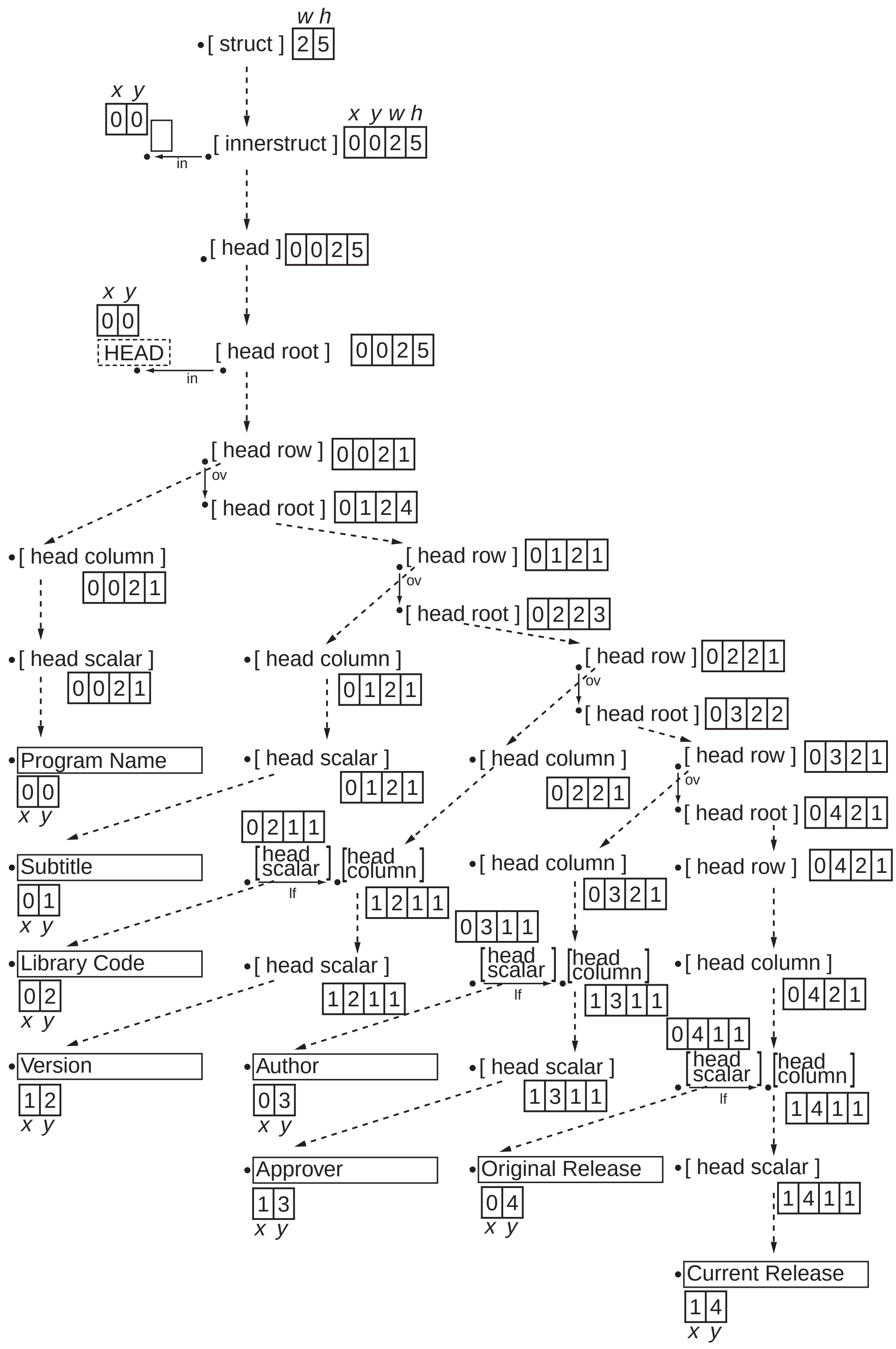


Fig. 3.1 The derivation tree for the Hiform2000 form shown in Fig. 2.2, where (1) *w* denotes an attribute *width*. (2) *h* denotes an attribute *height*.

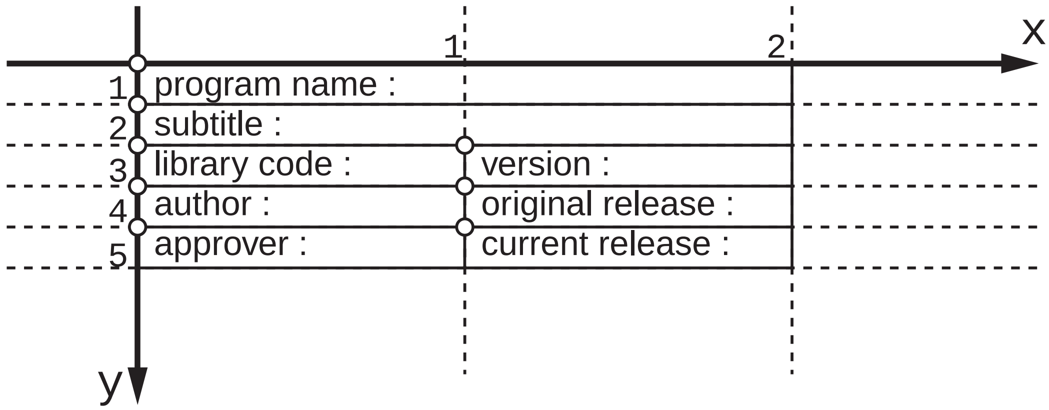


Fig. 3.2 Layout of cells in the form in Fig 2.2 by Fig 3.1.

Chapter 4

Tessellation Forms

We consider here tessellation forms that represent tables such as *symbol* tables. We note that ISO6592 does not issue about any symbol tables. We introduce an attribute NCE context-sensitive graph grammar that generates *tessellation forms*.

4.1 NCE Graph Grammars [9]

Σ is the alphabet of node labels. Γ is the alphabet of edge labels. The set of all (*concrete*) graphs over Σ and Γ is denoted $GR_{\Sigma,\Gamma}$

A *graph with (neighborhood controlled) embedding* over Σ and Γ is a pair (H, C) with $H \in GR_{\Sigma,\Gamma}$ and $C \subseteq \Sigma \times \Gamma \times \Gamma \times V_H \times \{in, out\}$. C is the *connection relation* of (H, C), and each element $(\sigma, \beta, \gamma, x, d)$ of C (with $\sigma \in \Sigma$, $\beta, \gamma \in \Gamma$, $x \in V_H$, and $d \in \{in, out\}$ is a *connection instruction* of (H, C)

The set of all graphs with embedding over Σ and Γ is denoted $GRE_{\Sigma,\Gamma}$.

Definition 4.1.1 [9]. An *edNCE grammar* is a tuple $G = (\Sigma, \Delta, \Gamma, \Omega, P, S)$, where Σ is the alphabet of node labels, $\Delta \subseteq \Sigma$ is the alphabet of terminal node labels, Γ is the alphabet of edge labels, $\Omega \subseteq \Gamma$ is the alphabet of final edge labels, P is the finite set of productions, and $S \in \Sigma - \Delta$ is the initial nonterminal. A production is of the form $X \rightarrow (D, C)$ with $X \in \Sigma - \Delta$ and $(D, C) \in GRE_{\Sigma,\Gamma}$. □

4.2 A Context-sensitive Attribute Graph Grammar for Tessellation Forms

We consider here an edNCE context-sensitive graph grammar for tessellation forms.

We extend edNCE graph grammars and introduce a context-sensitive attribute graph grammar.

Definition 4.2.1 *An attribute NCE graph grammar* is a 3-tuple $AGG = \langle G, Att, F \rangle$, where

1. $G = (\Sigma, \Delta, \Gamma, \Omega, P, S)$ is called an *underlying graph grammar* of AGG. Each production p in P is denoted by $p = X \to (D, C)$. $Lab(D)$ denotes the set of all occurrences of the node symbols labeling the nodes in the graph D.

2. Each node symbol $Y \in V$ of G has two disjoint finite sets $Inh(Y)$ and $Syn(Y)$ of *inherited* and *synthesized attributes*, respectively. We denote the set of all attributes of nonterminal node symbols X by $Att(Y) = Inh(Y) \cup Syn(Y)$. $Att = \bigcup_{Y \in V} Att(Y)$ is called the set of attributes of AGG. We assume that $Inh(S) = \emptyset$. An attribute a of Y is denoted by $a(Y)$, and set of possible values of a is denoted by $V(a)$.

3. Associated with each production $p = X_0 \to (D, C) \in P$ is a set F_p of *semantic rules* which define all the attributes in $Syn(X_0) \cup \bigcup_{Y \in Lab(D)} Inh(Y)$. A semantic rule defining an attribute $a_0(X_{i_0})$ has the form $a_0(X_{i_0}) := f(a_1(X_{i_1}),$ ..., $a_m(X_{i_m}))$, $0 \leq i_j \leq |Lab(D)|$, $X_{i_j} \in Lab(D), 0 \leq j \leq m$. Here $|Lab(D)|$ denotes the cardinality of the set $Lab(D)$, and f is a mapping from $V(a_1(X_{i_1}) \times$ ... $\times a_m(Xi_m))$ into $V(a_0(X_{i_0}))$. In this situation, we say that $a_0(X_{i_0})$ depends on $a_j(X_{i_j})$ for $j, 1 \leq j \leq m$ in p. The set $F = \bigcup_{p \in P} F_p$ is called the *set of semantic rules* of AGG.

□

We illustrate a tessellation form and its corresponding graph. We show productions of an attribute NCE graph grammar TFAGG in Appendix B for tessellation forms.

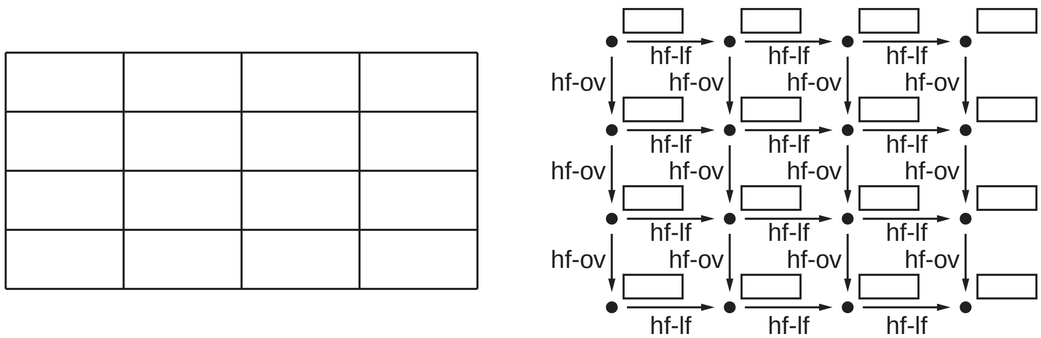


Fig. 4.1 A tessellation form and its corresponding graph.

Chapter 5

Conclusion

We proposed an attribute graph grammar that characterizes ISO6592 based program documentation forms with respect to both the logical and visual structures. The attribute graph grammar has 280 productions and 1248 attribute rules. And we showed that the attribute graph grammar is an attribute precedence graph grammar. Furthermore, we proposed an attribute graph grammar, which characterizes tessellation forms. This attribute graph grammar has 69 productions and 308 attribute rules.

We are now developing a software documentation system utilizing our proposed approach in this thesis.

References

[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] T. Nishino, Attribute Graph Grammars with Applications to Hichart Program Chart Editors., Advances in Software Science and Technology 1, 426-433 (1989)

[5] 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)

[6] K. Sugita, A. Adachi, Y. Miyadera, K. Tsuchida and T. Yaku, A visual programming environment based on graph grammars and tidy graph drawing, Proc. Internat. Conf. Software Engin. (ICSE '98) 20-II, 74-79 (1998)

[7] A. Adachi, T. Tsuchida and T. Yaku, Program visualization using attribute graph grammars, CD-ROM Book, IFIP World Computer Congress 98, (1998)

[8] Y. Adachi, K. Anzai, K. Tsuchida and T. Yaku, Hierarchical program diagram editor based on attribute graph grammar, Proc. IEEE COMPSAC 21, 205-213 (1996)

[9] Grsegorz Rozenberg (Ed.), Handbook of Graph Grammar and Computing by Graph Transformation,World Scientific Publishing(1997).

Appendix A
An Attribute Graph Grammar for Hi-form

No.	Production	Semantic rule
1	[struct]$_0$:= □$_1$ [innerstruct]$_2$	x(1)=0 y(1)=0 x(2)=x(1) y(2)=y(1) width(0) = width(2) height(0)= height(2)
2	[innerstruct]$_0$:= [head]$_1$ [body]$_2$	x(1)=x(0) + Mleft y(1)=y(0) + Mtop x(2)=x(1) + Mleft y(2)=y(1) +height(1)+Mcen width(0) = max(width(1),width(2)) height(0)= height(1)+height(2) +Mtop+Mcen+Mbottom
H1	[head]$_0$:= HEAD$_1$ [head root]$_2$	x(1)=0 y(1)=0 x(2)=x(1) + HMleft y(2)=y(1) + HMtop width(0) = width(2) +HMleft+HMright height(0)=height(2) +HMtop+HMbottom
H2	[head root]$_0$:= [head row]$_1$ [head root]$_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) y(2)=y(1) +height(1)+HSv width(0) = max(width(1),width(2)) height(0)= height(1)+height(2) +HSv
H3	[head root]$_0$:= [head row]$_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
H4	[head row]$_0$:= [head column]$_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
H5	[head column]$_0$:= [head scalar]$_1$ [head column]$_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) + width(1) + HSh y(2)=y(1) width(0) = width(1)+width(2)+HSh height(0)= max(height(1),height(2))
H6	[head column]$_0$:= [head scalar]$_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
H7	[head scalar]$_0$:= Program Name$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_pname height(0)= HEIGHT_pneme
H8	[head scalar]$_0$:= Subtitle$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_stitle height(0)= HEIGHT_stitle
H9	[head scalar]$_0$:= Library Code$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_lcode height(0)= HEIGHT_lcode
H10	[head scalar]$_0$:= Version$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_version height(0)= HEIGHT_version
H11	[head scalar]$_0$:= Author$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_author height(0)= HEIGHT_author
H12	[head scalar]$_0$:= Approver$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_approver height(0)= HEIGHT_approver
H13	[head scalar]$_0$:= Original Release$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_orelease height(0)= HEIGHT_orelease
H14	[head scalar]$_0$:= Current Release$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_crelease height(0)= HEIGHT_crelease

No.	Production	Semantic rule
A1-0	[body]$_0$:= [a1]$_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
A1-1	[a1]$_0$:= A1$_1$ [a1 root]$_2$	x(1)=0 y(1)=0 x(2)=x(1)+A1Mleft y(2)=y(1)+A1Mtop width(0) = width(2) +A1Mleft+A1Mright height(0)=height(2) +A1Mtop+A1Mbottom
A1-2	[a1 root]$_0$:= [a1 row]$_1$ [a1 root]$_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) y(2)=y(1) + height(1) + A1Sv width(0) = max(width(1),width(2)) height(0)= height(1)+height(2) +A1Sv
A1-3	[a1 root]$_0$:= [a1 row]$_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
A1-4	[a1 row]$_0$:= [a1 column]$_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
A1-5	[a1 column]$_0$:= [a1 scalar]$_1$ [a1 column]$_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) +width(1)+A1Sh y(2)=y(1) width(0) = width(1)+width(2)+A1Sh height(0)= max(height(1),height(2))
A1-6	[a1 column]$_0$:= [a1 scalar]$_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
A1-7	[a1 scalar]$_0$:= Key Words$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_keyword height(0)= HEIGHT_keyword
A1-8	[a1 scalar]$_0$:= CR Code$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_crcode height(0)= HEIGHT_crcode
A1-9	[a1 scalar]$_0$:= Scope$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_scape height(0)= HEIGHT_scape
A1-10	[a1 scalar]$_0$:= Variant$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_variant height(0)= HEIGHT_variant
A1-11	[a1 scalar]$_0$:= Language$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_language height(0)= HEIGHT_language
A1-12	[a1 scalar]$_0$:= Operation$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_operarion height(0)= HEIGHT_operation
A1-13	[a1 scalar]$_0$:= Software Req.$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_softreq height(0)= HEIGHT_softreq
A1-14	[a1 scalar]$_0$:= Hardware Req.$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_hardreq height(0)= HEIGHT_hardreq
A1-15	[a1 scalar]$_0$:= References$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_reference height(0)= HEIGHT_reference
A1-16	[a1 scalar]$_0$:= Function$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_function height(0)= HEIGHT_function
A1-17	[a1 scalar]$_0$:= Example$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_example height(0)= HEIGHT_example

No.	Production	Semantic rule
A2-0	[body]$_0$:= [a2]$_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
A2-1	[a2]$_0$:= A2$_1$ [a1 root]$_2$	x(1)=0 y(1)=0 x(2)=x(1)+A2Mleft y(2)=y(1)+A2Mtop width(0) = width(2) +A2Mleft+A2Mright height(0)=height(2) +A2Mtop+A2Mbottom
A2-2	[a2 root]$_0$:= [a2 row]$_1$ [a2 root]$_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) y(2)=y(1) + height(1) + A2Sv width(0) = max(width(1),width(2)) height(0)= height(1)+height(2) +A2Sv
A2-3	[a2 root]$_0$:= [a2 row]$_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
A2-4	[a2 row]$_0$:= [a2 column]$_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
A2-5	[a1 column]$_0$:= [a2 scalar]$_1$ [a2 column]$_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) +width(1)+A2Sh y(2)=y(1) width(0) = width(1)+width(2)+A2Sh height(0)= max(height(1),height(2))
A2-6	[a2 column]$_0$:= [a2 scalar]$_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
A2-7	[a2 scalar]$_0$:= Hystory$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_hystory height(0)= HEIGHT_hystory
A2-8	[a2 scalar]$_0$:= Responsibility$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_respons height(0)= HEIGHT_respons
A2-9	[a2 scalar]$_0$:= Data Prt. & Scr.$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_dpc height(0)= HEIGHT_dpc
A2-10	[a2 scalar]$_0$:= Ope. Con. Inst.$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_opeci height(0)= HEIGHT_opeci
A2-11	[a2 scalar]$_0$:= Ope. Message$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_opem height(0)= HEIGHT_opem
A2-12	[a2 scalar]$_0$:= Instal. & Support$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_instsupp height(0)= HEIGHT_instsupp

No.	Production	Semantic rule
A3-0	[body]$_0$:= [a3]$_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
A3-1	[a3]$_0$:= A3$_1$ [a3 root]$_2$	x(1)=0 y(1)=0 x(2)=x(1)+A3Mleft y(2)=y(1)+A3Mtop width(0) = width(2) +A3Mleft+A3Mright height(0)=height(2) +A3Mtop+A3Mbottom
A3-2	[a3 root]$_0$:= [a3 row]$_1$ [a3 root]$_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) y(2)=y(1) + height(1) + A3Sv width(0) = max(width(1),width(2)) height(0)= height(1)+height(2) +A3Sv
A3-3	[a3 root]$_0$:= [a3 row]$_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
A3-4	[a3 row]$_0$:= [a3 column]$_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
A3-5	[a3 column]$_0$:= [a3 scalar]$_1$ [a3 column]$_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) +width(1)+A3Sh y(2)=y(1) width(0) = width(1)+width(2)+A3Sh height(0)= max(height(1),height(2))
A3-6	[a3 column]$_0$:= [a3 scalar]$_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
A3-7	[a3 scalar]$_0$:= Legal Conditions$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_legalcond height(0)= HEIGHT_legalcond
A3-8	[a3 scalar]$_0$:= Price$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_price height(0)= HEIGHT_price
A3-9	[a3 scalar]$_0$:= Installation$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_inst height(0)= HEIGHT_inst
A3-10	[a3 scalar]$_0$:= Training$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_training height(0)= HEIGHT_training
A3-11	[a3 scalar]$_0$:= Maintenance$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_maint height(0)= HEIGHT_maint
A3-12	[a3 scalar]$_0$:= Quality Assurance$_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_qassur height(0)= HEIGHT_qassur

No.	Production	Semantic rule
A4-0	ov [body]$_0$:= [a4]$_1$ in	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
A4-1	ov [a4]$_0$:= ov A4 $_1$ [a4 root]$_2$ in,ov in in,ov	x(1)=0 y(1)=0 x(2)=x(1)+A4Mleft y(2)=y(1)+A4Mtop width(0) = width(2) +A4Mleft+A4Mright height(0)=height(2) +A4Mtop+A4Mbottom
A4-2	[a4 root]$_0$:= ov [a4 row]$_1$ ov [a4 root]$_2$ in in	x(1)=x(0) y(1)=y(0) x(2)=x(1) y(2)=y(1) + height(1) + A4Sv width(0) = max(width(1),width(2)) height(0)= height(1)+height(2) +A4Sv
A4-3	ov [a4 root]$_0$:= [a4 row]$_1$ in	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
A4-4	ov [a4 row]$_0$:= [a4 column]$_1$ in,ov	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
A4-5	ov,lf [a4 column]$_0$:= ov,lf [a4 scalar]$_1$ [a4 column]$_2$ in,ov lf	x(1)=x(0) y(1)=y(0) x(2)=x(1) +width(1)+A4Sh y(2)=y(1) width(0) = width(1)+width(2)+A4Sh height(0)= max(height(1),height(2))
A4-6	ov,lf [a4 column]$_0$:= [a4 scalar]$_1$ in,ov	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
A4-7	ov,lf [a4 scalar]$_0$:= Identifier Name $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_idname height(0)= HEIGHT_idname
A4-8	ov,lf [a4 scalar]$_0$:= Identifier Category $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_idcategory height(0)= HEIGHT_idcategory
A4-9	ov,lf [a4 scalar]$_0$:= Purpose $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_purpose height(0)= HEIGHT_purpose
A4-10	ov,lf [a4 scalar]$_0$:= Value / Range $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_valuerange height(0)= HEIGHT_valuerange
A4-11	ov,lf [a4 scalar]$_0$:= Unit $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_unit height(0)= HEIGHT_unit
A4-12	ov,lf [a4 scalar]$_0$:= Rest. $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_rest height(0)= HEIGHT_rest
A4-13	ov,lf [a4 scalar]$_0$:= Ref. $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_ref height(0)= HEIGHT_ref

No.	Production	Semantic rule
A5-0	ov [body]$_0$:= [a5]$_1$ in	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
A5-1	ov [a5]$_0$:= ov A5 $_1$ [a5 root]$_2$ in,ov in in,ov	x(1)=0 y(1)=0 x(2)=x(1)+A5Mleft y(2)=y(1)+A5Mtop width(0) = width(2) +A5Mleft+A5Mright height(0)=height(2) +A5Mtop+A5Mbottom
A5-2	[a5 root]$_0$:= ov [a5 row]$_1$ ov [a5 root]$_2$ in in	x(1)=x(0) y(1)=y(0) x(2)=x(1) y(2)=y(1) + height(1) + A5Sv width(0) = max(width(1),width(2)) height(0)= height(1)+height(2) +A5Sv
A5-3	ov [a5 root]$_0$:= [a5 row]$_1$ in	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
A5-4	ov [a5 row]$_0$:= [a5 column]$_1$ in,ov	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
A5-5	ov,lf [a5 column]$_0$:= ov,lf [a5 scalar]$_1$ [a5 column]$_2$ in,ov lf	x(1)=x(0) y(1)=y(0) x(2)=x(1) +width(1)+A5Sh y(2)=y(1) width(0) = width(1)+width(2)+A5Sh height(0)= max(height(1),height(2))
A5-6	ov,lf [a5 column]$_0$:= [a5 scalar]$_1$ in,ov	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
A5-7	ov,lf [a5 scalar]$_0$:= Prob. Descript. $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_pdescript. height(0)= HEIGHT_pdescript
A5-8	ov,lf [a5 scalar]$_0$:= Prob. Suppl. Info. $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_psupplinfo height(0)= HEIGHT_psupplinfo
A5-9	ov,lf [a5 scalar]$_0$:= Prob. Solution $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_psolution height(0)= HEIGHT_psolution

No.	Production	Semantic rule	
A6-0	ov [body]0 in := ov [a6]1 in	x(1)=x(0) y(1)=y(0)	width(0) = width(1) height(0)= height(1)
A6-1	ov [a6]0 in,ov := ov A6 1 in [a6 root]2 in,ov	x(1)=0 y(1)=0 x(2)=x(1)+A6Mleft y(2)=y(1)+A6Mtop	width(0) = width(2) +A6Mleft+A6Mright height(0)=height(2) +A6Mtop+A6Mbottom
A6-2	[a6 root]0 in := ov [a6 row]1 in ov [a6 root]2	x(1)=x(0) y(1)=y(0) x(2)=x(1) y(2)=y(1) + height(1) + A6Sv	width(0) = max(width(1),width(2)) height(0)= height(1)+height(2) +A6Sv
A6-3	ov [a6 root]0 in := ov [a6 row]1 in	x(1)=x(0) y(1)=y(0)	width(0) = width(1) height(0)= height(1)
A6-4	ov [a6 row]0 in,ov := ov [a6 column]1 in,ov	x(1)=x(0) y(1)=y(0)	width(0) = width(1) height(0)= height(1)
A6-5	ov,lf [a6 column]0 in,ov := ov,lf [a6 scalar]1 lf [a6 column]2 in,ov	x(1)=x(0) y(1)=y(0) x(2)=x(1) +width(1)+A6Sh y(2)=y(1)	width(0) = width(1)+width(2)+A6Sh height(0)= max(height(1),height(2))
A6-6	ov,lf [a6 column]0 in,ov := ov,lf [a6 scalar]1 in,ov	x(1)=x(0) y(1)=y(0)	width(0) = width(1) height(0)= height(1)
A6-7	ov,lf [a6 scalar]0 in,ov,lf := ov,lf [Functional Spe.]1 in,ov,lf	x(1)=x(0) y(1)=y(0)	width(0) = WIDTH_funcspe height(0)= HEIGHT_funcspe

No.	Production	Semantic rule	
B-1	ov [body]0 in := ov [b]1 in	x(1)=x(0) y(1)=y(0)	width(0) = width(1) height(0)= height(1)
B-2	ov [b]0 in,ov := ov B 1 in [bc body]1 in,ov ov [b body]2	x(1)=0 y(1)=0 x(2)=x(1)+BCMleft y(2)=y(1)+BCMtop	width(0) = width(2) +BCMleft+BCMright height(0)=height(2) +BCMtop+BCMbottom
BC-1	[bc body]0 in,ov := BC 1 in [bc root]2 in,ov	x(1)=0 y(1)=0 x(2)=x(1)+BCMleft y(2)=y(1)+BCMtop	width(0) = width(2) +BCMleft+BCMright height(0)=height(2) +BCMtop+BCMbottom
BC-2	[bc root]0 in := ov [bc row]1 in ov [bc root]2	x(1)=x(0) y(1)=y(0) x(2)=x(1) y(2)=y(1) + height(1) + BCSv	width(0) = max(width(1),width(2)) height(0)= height(1)+height(2) +BCSv
BC-3	ov [bc root]0 in := ov [bc row]1 in	x(1)=x(0) y(1)=y(0)	width(0) = width(1) height(0)= height(1)
BC-4	ov [bc row]0 in,ov := ov [bc column]1 in,ov	x(1)=x(0) y(1)=y(0)	width(0) = width(1) height(0)= height(1)
BC-5	ov,lf [bc column]0 in,ov := ov,lf [bc scalar]1 lf [bc column]2 in,ov	x(1)=x(0) y(1)=y(0) x(2)=x(1) +width(1)+BCSh y(2)=y(1)	width(0) = width(1)+width(2)+BCSh height(0)= max(height(1),height(2))
BC-6	ov,lf [bc column]0 in,ov := ov,lf [bc scalar]1 in,ov	x(1)=x(0) y(1)=y(0)	width(0) = width(1) height(0)= height(1)
BC-7	ov,lf [bc scalar]0 in,ov,lf := ov,lf [Technical Id.]1 in,ov,lf	x(1)=x(0) y(1)=y(0)	width(0) = WIDTH_technicalid height(0)= HEIGHT_technicalid
BC-8	ov,lf [bc scalar]0 in,ov,lf := ov,lf [Appl. Orient. Id.]1 in,ov,lf	x(1)=x(0) y(1)=y(0)	width(0) = WIDTH_aoid height(0)= HEIGHT_aoid

No.	Production	Semantic rule	
B1-0	ov [b]0 in := ov [b1]1 in	x(1)=x(0) y(1)=y(0)	width(0) = width(1) height(0)= height(1)
B1-1	ov [b1]0 in,ov := ov B1 1 in [b1 root]2 in,ov	x(1)=0 y(1)=0 x(2)=x(1)+B1Mleft y(2)=y(1)+B1Mtop	width(0) = width(2) +B1Mleft+B1Mright height(0)=height(2) +B1Mtop+B1Mbottom
B1-2	[b1 root]0 in := ov [b1 row]1 in ov [b1 root]2	x(1)=x(0) y(1)=y(0) x(2)=x(1) y(2)=y(1) + height(1) + B1Sv	width(0) = max(width(1),width(2)) height(0)= height(1)+height(2) +B1Sv
B1-3	ov [b1 root]0 in := ov [b1 row]1 in	x(1)=x(0) y(1)=y(0)	width(0) = width(1) height(0)= height(1)
B1-4	ov [b1 row]0 in,ov := ov [b1 column]1 in,ov	x(1)=x(0) y(1)=y(0)	width(0) = width(1) height(0)= height(1)
B1-5	ov,lf [b1 column]0 in,ov := ov,lf [b1 scalar]1 lf [b1 column]2 in,ov	x(1)=x(0) y(1)=y(0) x(2)=x(1) +width(1)+B1Sh y(2)=y(1)	width(0) = width(1)+width(2)+B1Sh height(0)= max(height(1),height(2))
B1-6	ov,lf [b1 column]0 in,ov := ov,lf [b1 scalar]1 in,ov	x(1)=x(0) y(1)=y(0)	width(0) = width(1) height(0)= height(1)
B1-7	ov,lf [b1 scalar]0 in,ov,lf := ov,lf [D. of Data Doc.]1 in,ov,lf	x(1)=x(0) y(1)=y(0)	width(0) = WIDTH_ddatadoc height(0)= HEIGHT_ddatadoc

No.	Production	Semantic rule
B2-0	[b body]$_0$ (ov; in) := [b2]$_1$ (ov; in)	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
B2-1	[b2]$_0$ (ov; in,ov) := B2 (ov; in) $_1$ [b2 root]$_2$	x(1)=0 y(1)=0 x(2)=x(1)+B2Mleft y(2)=y(1)+B2Mtop width(0) = width(2) +B2Mleft+B2Mright height(0)=height(2) +B2Mtop+B2Mbottom
B2-2	[b2 root]$_0$ (in) := [b2 row]$_1$ (ov; in) [b2 root]$_2$ (ov)	x(1)=x(0) y(1)=y(0) x(2)=x(1) y(2)=y(1) + height(1) + B2Sv width(0) = max(width(1),width(2)) height(0)= height(1)+height(2) +B2Sv
B2-3	[b2 root]$_0$ (ov; in) := [b2 row]$_1$ (ov; in)	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
B2-4	[b2 row]$_0$ (ov; in,ov) := [b2 column]$_1$ (ov; in,ov)	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
B2-5	[b2 column]$_0$ (ov,lf; in,ov) := [b2 scalar]$_1$ (ov,lf; in,ov) (lf) [b2 column]$_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) +width(1)+B2Sh y(2)=y(1) width(0) = width(1)+width(2)+B2Sh height(0)= max(height(1),height(2))
B2-6	[b2 column]$_0$ (ov,lf; in,ov) := [b2 scalar]$_1$ (ov,lf; in,ov)	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
B2-7	[b2 scalar]$_0$ (ov,lf; in,ov,lf) := Valid through $_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_vthrought height(0)= HEIGHT_vthrought
B2-8	[b2 scalar]$_0$ (ov,lf; in,ov,lf) := Variants $_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_variant height(0)= HEIGHT_variant
B2-9	[b2 scalar]$_0$ (ov,lf; in,ov,lf) := Validity $_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_validity height(0)= HEIGHT_validity
B2-10	[b2 scalar]$_0$ (ov,lf; in,ov,lf) := Access Author. $_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_aauthor height(0)= HEIGHT_aauthor
B2-11	[b2 scalar]$_0$ (ov,lf; in,ov,lf) := Origination $_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_origination height(0)= HEIGHT_origination
B2-12	[b2 scalar]$_0$ (ov,lf; in,ov,lf) := Read Access $_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_readaccess height(0)= HEIGHT_readaccess
B2-13	[b2 scalar]$_0$ (ov,lf; in,ov,lf) := Amentdment $_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_amentd height(0)= HEIGHT_amentd
B2-14	[b2 scalar]$_0$ (ov,lf; in,ov,lf) := Communication $_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_communication height(0)= HEIGHT_communication
B2-15	[b2 scalar]$_0$ (ov,lf; in,ov,lf) := Access Regulation $_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_areculation height(0)= HEIGHT_aregulation
B2-16	[b2 scalar]$_0$ (ov,lf; in,ov,lf) := Responsibilities $_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_responsibilities height(0)= HEIGHT_responsibilities
B2-17	[b2 scalar]$_0$ (ov,lf; in,ov,lf) := Appli. Oriented $_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_aoriented height(0)= HEIGHT_aoriented

No.	Production	Semantic rule
B2-18	[b2 scalar]$_0$ (ov,lf; in,ov,lf) := Organizational $_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_organizational height(0)= HEIGHT_organizational
B2-19	[b2 scalar]$_0$ (ov,lf; in,ov,lf) := Technical $_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_technical height(0)= HEIGHT_technical
B2-20	[b2 scalar]$_0$ (ov,lf; in,ov,lf) := Custodial $_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_custodial height(0)= HEIGHT_custodial
B2-21	[b2 scalar]$_0$ (ov,lf; in,ov,lf) := Data Security $_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_dsecurity height(0)= HEIGHT_dsecurity
B2-22	[b2 scalar]$_0$ (ov,lf; in,ov,lf) := Recovery $_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_recovery height(0)= HEIGHT_recovery
B2-23	[b2 scalar]$_0$ (ov,lf; in,ov,lf) := Encryption $_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_encryption height(0)= HEIGHT_encryption
B2-24	[b2 scalar]$_0$ (ov,lf; in,ov,lf) := Use $_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_use height(0)= HEIGHT_use

No.	Production	Semantic rule
B3-0	ov [b body]$_0$ in := ov [b3]$_1$ in	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
B3-1	ov [b3]$_0$ in,ov := ov B3 $_1$ in [b3 root]$_2$ in,ov	x(1)=0 y(1)=0 x(2)=x(1)+B3Mleft y(2)=y(1)+B3Mtop width(0) = width(2) +B3Mleft+B3Mright height(0)=height(2) +B3Mtop+B3Mbottom
B3-2	[b3 root]$_0$ in := ov [b3 row]$_1$ in ov [b3 root]$_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) y(2)=y(1) + height(1) + B3Sv width(0) = max(width(1),width(2)) height(0)= height(1)+height(2) +B3Sv
B3-3	ov [b3 root]$_0$ in := ov [b3 row]$_1$ in	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
B3-4	ov [b3 row]$_0$ in,ov := ov [b3 column]$_1$ in,ov	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
B3-5	ov,lf [b3 column]$_0$ in,ov := ov,lf [b3 scalar]$_1$ in,ov lf [b3 column]$_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) +width(1)+B3Sh y(2)=y(1) width(0) = width(1)+width(2)+B3Sh height(0)= max(height(1),height(2))
B3-6	ov,lf [b3 column]$_0$ in,ov := ov,lf [b3 scalar]$_1$ in,ov	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
B3-7	ov,lf [b3 scalar]$_0$ in,ov,lf := ov,lf Category $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_category height(0)= HEIGHT_category
B3-8	ov,lf [b3 scalar]$_0$ in,ov,lf := ov,lf Status $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_status height(0)= HEIGHT_status
B3-9	ov,lf [b3 scalar]$_0$ in,ov,lf := ov,lf Purpose $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_purpose height(0)= HEIGHT_purpose
B3-10	ov,lf [b3 scalar]$_0$ in,ov,lf := ov,lf Descriptors $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_descriptor height(0)= HEIGHT_descriptor
B3-11	ov,lf [b3 scalar]$_0$ in,ov,lf := ov,lf Sensitivity $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_sensitivity height(0)= HEIGHT_sensitivity
B3-12	ov,lf [b3 scalar]$_0$ in,ov,lf := ov,lf Format $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_format height(0)= HEIGHT_format
B3-13	ov,lf [b3 scalar]$_0$ in,ov,lf := ov,lf Size $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_size height(0)= HEIGHT_size
B3-14	ov,lf [b3 scalar]$_0$ in,ov,lf := ov,lf Medium $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_medium height(0)= HEIGHT_medium
B3-15	ov,lf [b3 scalar]$_0$ in,ov,lf := ov,lf Compression $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_compression height(0)= HEIGHT_compression
B3-16	ov,lf [b3 scalar]$_0$ in,ov,lf := ov,lf Code $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_code height(0)= HEIGHT_code
B3-17	ov,lf [b3 scalar]$_0$ in,ov,lf := ov,lf Character Set $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_characterset height(0)= HEIGHT_characterset

No.	Production	Semantic rule
B3-18	ov,lf [b3 scalar]$_0$ in,ov,lf := ov,lf Data Type $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_datatype height(0)= HEIGHT_datatype
B3-19	ov,lf [b3 scalar]$_0$ in,ov,lf := ov,lf Units $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_unit height(0)= HEIGHT_unit
B3-20	ov,lf [b3 scalar]$_0$ in,ov,lf := ov,lf Range of Values $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_rangeofvalue height(0)= HEIGHT_rangeofvalue
B3-21	ov,lf [b3 scalar]$_0$ in,ov,lf := ov,lf Encoding $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_encoding height(0)= HEIGHT_encoding
B3-22	ov,lf [b3 scalar]$_0$ in,ov,lf := ov,lf Cheking Condition $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_chekingcondition height(0)= HEIGHT_chekingcondition
B3-23	ov,lf [b3 scalar]$_0$ in,ov,lf := ov,lf Occurrence $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_occurrence height(0)= HEIGHT_occurrence
B3-24	ov,lf [b3 scalar]$_0$ in,ov,lf := ov,lf Dependencies $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_dependency height(0)= HEIGHT_dependency

No.	Production	Semantic rule
C-1	ov [body]$_0$ in := ov [c]$_1$ in	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
C-2	ov [c]$_0$ in,ov := ov C$_1$ in, in,ov [cc body]$_1$ ov [c body]$_2$	x(1)=0 y(1)=0 x(2)=x(1)+CCMleft y(2)=y(1)+CCMtop width(0) = width(2) +CCMleft+CCMright height(0)=height(2) +CCMtop+CCMbottom
CC-1	[cc body]$_0$ in,ov := CC$_1$ in, in,ov [cc root]$_2$	x(1)=0 y(1)=0 x(2)=x(1)+CCMleft y(2)=y(1)+CCMtop width(0) = width(2) +CCMleft+CCMright height(0)=height(2) +CCMtop+CCMbottom
CC-2	[bc root]$_0$ in := ov [cc row]$_1$ in, ov [cc root]$_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) y(2)=y(1) + height(1) + CCSv width(0) = max(width(1),width(2)) height(0)= height(1)+height(2) +CCSv
CC-3	ov [cc root]$_0$ in := ov [cc row]$_1$ in	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
CC-4	ov [cc row]$_0$ in,ov := ov [cc column]$_1$ in,ov	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
CC-5	ov,lf [cc column]$_0$ in,ov := ov,lf [cc scalar]$_1$ in,ov lf [cc column]$_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) +width(1)+CCSh y(2)=y(1) width(0) = width(1)+width(2)+CCSh height(0)= max(height(1),height(2))
CC-6	ov,lf [cc column]$_0$ in,ov := ov,lf [cc scalar]$_1$ in,ov	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
CC-7	ov,lf [cc scalar]$_0$ in,ov,lf := ov,lf [Procedure name]$_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_procedurename height(0)= HEIGHT_procedurename
CC-8	ov,lf [cc scalar]$_0$ in,ov,lf := ov,lf [Variants]$_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_variant height(0)= HEIGHT_variant

No.	Production	Semantic rule
C1-0	ov [c]$_0$ in := ov [c1]$_1$ in	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
C1-1	ov [c1]$_0$ in,ov := ov C1$_1$ in, in,ov [c1 root]$_2$	x(1)=0 y(1)=0 x(2)=x(1)+C1Mleft y(2)=y(1)+C1Mtop width(0) = width(2) +C1Mleft+C1Mright height(0)=height(2) +C1Mtop+C1Mbottom
C1-2	[c1 root]$_0$ in := ov [c1 row]$_1$ in, ov [c1 root]$_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) y(2)=y(1) + height(1) + C1Sv width(0) = max(width(1),width(2)) height(0)= height(1)+height(2) +C1Sv
C1-3	ov [c1 root]$_0$ in := ov [c1 row]$_1$ in	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
C1-4	ov [c1 row]$_0$ in,ov := ov [c1 column]$_1$ in,ov	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
C1-5	ov,lf [c1 column]$_0$ in,ov := ov,lf [c1 scalar]$_1$ in,ov lf [c1 column]$_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) +width(1)+C1Sh y(2)=y(1) width(0) = width(1)+width(2)+C1Sh height(0)= max(height(1),height(2))
C1-6	ov,lf [c1 column]$_0$ in,ov := ov,lf [c1 scalar]$_1$ in,ov	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
C1-7	ov,lf [c1 scalar]$_0$ in,ov,lf := ov,lf [D. of Procedure]$_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_dprocedure height(0)= HEIGHT_dprocedure

No.	Production	Semantic rule
C2-0	ov [c body]$_0$ in := ov [c2]$_1$ in	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
C2-1	ov [c2]$_0$ in,ov := ov C2 $_1$ in [c2 root]$_2$ in,ov	x(1)=0 y(1)=0 x(2)=x(1)+C2Mleft y(2)=y(1)+C2Mtop width(0) = width(2) +C2Mleft+C2Mright height(0)=height(2) +C2Mtop+C2Mbottom
C2-2	[c2 root]$_0$ in := ov [c2 row]$_1$ in ov [c2 root]$_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) y(2)=y(1) + height(1) + C2Sv width(0) = max(width(1),width(2)) height(0)= height(1)+height(2) +C2Sv
C2-3	ov [c2 root]$_0$ in := ov [c2 row]$_1$ in	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
C2-4	ov [c2 row]$_0$ in,ov := ov [c2 column]$_1$ in,ov	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
C2-5	ov,lf [c2 column]$_0$ in,ov := ov,lf [c2 scalar]$_1$ lf [c2 column]$_2$ in,ov	x(1)=x(0) y(1)=y(0) x(2)=x(1) +width(1)+C2Sh y(2)=y(1) width(0) = width(1)+width(2)+C2Sh height(0)= max(height(1),height(2))
C2-6	ov,lf [c2 column]$_0$ in,ov := ov,lf [c2 scalar]$_1$ in,ov	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
C2-7	ov,lf [c2 scalar]$_0$ in,ov,lf := ov,lf Responsibilities $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_responsibility height(0)= HEIGHT_responsibility
C2-8	ov,lf [c2 scalar]$_0$ in,ov,lf := ov,lf Development $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_developmant height(0)= HEIGHT_development
C2-9	ov,lf [c2 scalar]$_0$ in,ov,lf := ov,lf Destribution $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_destribution height(0)= HEIGHT_destribution
C2-10	ov,lf [c2 scalar]$_0$ in,ov,lf := ov,lf Training $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_training height(0)= HEIGHT_training
C2-11	ov,lf [c2 scalar]$_0$ in,ov,lf := ov,lf Modification $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_modification height(0)= HEIGHT_modification
C2-12	ov,lf [c2 scalar]$_0$ in,ov,lf := ov,lf Contractual Items $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_contractualitem height(0)= HEIGHT_contractualitem
C2-13	ov,lf [c2 scalar]$_0$ in,ov,lf := ov,lf Legal Condition $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_lagalcondition height(0)= HEIGHT_legalcondition
C2-14	ov,lf [c2 scalar]$_0$ in,ov,lf := ov,lf Training $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_training height(0)= HEIGHT_training
C2-15	ov,lf [c2 scalar]$_0$ in,ov,lf := ov,lf Quality Assurance $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_qualityassurance height(0)= HEIGHT_qualityassurance
C2-16	ov,lf [c2 scalar]$_0$ in,ov,lf := ov,lf Maintenance $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_maintenance height(0)= HEIGHT_maintenance
C2-17	ov,lf [c2 scalar]$_0$ in,ov,lf := ov,lf Destri. & Filing $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_destribtutionfiling height(0)= HEIGHT_destributionfiling

No.	Production	Semantic rule
C2-18	ov,lf [c2 scalar]$_0$ in,ov,lf := ov,lf Testing $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_testing height(0)= HEIGHT_testing
C2-19	ov,lf [c2 scalar]$_0$ in,ov,lf := ov,lf Training $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_training height(0)= HEIGHT_training
C2-20	ov,lf [c2 scalar]$_0$ in,ov,lf := ov,lf Refinement Ref. $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_refinementrefer height(0)= HEIGHT_refinementrefer
C2-21	ov,lf [c2 scalar]$_0$ in,ov,lf := ov,lf Adapt. Suggestion $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_adaptaionsuggest height(0)= HEIGHT_adaptationsuggest
C2-22	ov,lf [c2 scalar]$_0$ in,ov,lf := ov,lf Supp. Procedure $_1$ in,ov,lf	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_supportofprocedure height(0)= HEIGHT_supportofprocedure

No.	Production	Semantic rule
C3-0	[c body]$_0$ (ov; in) := [c3]$_1$ (ov; in)	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
C3-1	[c3]$_0$ (ov; in,ov) := C3$_1$ (ov; in,ov) →in [c3 root]$_2$	x(1)=0 y(1)=0 x(2)=x(1)+C3Mleft y(2)=y(1)+C3Mtop width(0) = width(2) +C3Mleft+C3Mright height(0)=height(2) +C3Mtop+C3Mbottom
C3-2	[c3 root]$_0$ (in) := [c3 row]$_1$ (ov; in) [c3 root]$_2$ (ov)	x(1)=x(0) y(1)=y(0) x(2)=x(1) y(2)=y(1) + height(1) + C3Sv width(0) = max(width(1),width(2)) height(0)= height(1)+height(2) +C3Sv
C3-3	[c3 root]$_0$ (ov; in) := [c3 row]$_1$ (ov; in)	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
C3-4	[c3 row]$_0$ (ov; in,ov) := [c3 column]$_1$ (ov; in,ov)	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
C3-5	[c3 column]$_0$ (ov,lf; in,ov) := [c3 scalar]$_1$ (ov,lf; in,ov) →lf [c3 column]$_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) +width(1)+C3Sh y(2)=y(1) width(0) = width(1)+width(2)+C3Sh height(0)= max(height(1),height(2))
C3-6	[c3 column]$_0$ (ov,lf; in,ov) := [c3 scalar]$_1$ (ov,lf; in,ov)	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
C3-7	[c3 scalar]$_0$ (ov,lf; in,ov,lf) := References$_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_reference height(0)= HEIGHT_reference
C3-8	[c3 scalar]$_0$ (ov,lf; in,ov,lf) := Occ. Frequency$_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_occfrequency height(0)= HEIGHT_occfrecuency
C3-9	[c3 scalar]$_0$ (ov,lf; in,ov,lf) := Function$_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_function height(0)= HEIGHT_function
C3-10	[c3 scalar]$_0$ (ov,lf; in,ov,lf) := Cap. & R. Req.$_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_capabilityrreq height(0)= HEIGHT_capabilityrreq
C3-11	[c3 scalar]$_0$ (ov,lf; in,ov,lf) := Rest. & Excep.$_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_restexception height(0)= HEIGHT_restexception
C3-12	[c3 scalar]$_0$ (ov,lf; in,ov,lf) := Personnel$_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_personnel height(0)= HEIGHT_personnel
C3-13	[c3 scalar]$_0$ (ov,lf; in,ov,lf) := Data Prtc. & Sec.$_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_dataprtcsec height(0)= HEIGHT_dataprtcsec
C3-14	[c3 scalar]$_0$ (ov,lf; in,ov,lf) := Personnel Skill$_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_personnelskill height(0)= HEIGHT_personnelskill
C3-15	[c3 scalar]$_0$ (ov,lf; in,ov,lf) := Hardware Req.$_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_hardwarereq height(0)= HEIGHT_hardwarereq
C3-16	[c3 scalar]$_0$ (ov,lf; in,ov,lf) := Software Req.$_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_softwarereq height(0)= HEIGHT_softwarereq
C3-17	[c3 scalar]$_0$ (ov,lf; in,ov,lf) := Supplies$_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_supplies height(0)= HEIGHT_supplies

No.	Production	Semantic rule
C3-18	[c3 scalar]$_0$ (ov,lf; in,ov,lf) := Timing Constraints$_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_tconstraint height(0)= HEIGHT_tconstraint
C3-19	[c3 scalar]$_0$ (ov,lf; in,ov,lf) := Associated Doc.$_1$ (ov,lf; in,ov,lf)	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_associateddoc height(0)= HEIGHT_associateddoc

No.	Production	Semantic rule
C4-0	$[\text{c body}]_0$:= $[\text{c4}]_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
C4-1	$[\text{c4}]_0$:= C4_1 $[\text{c4 root}]_2$	x(1)=0 y(1)=0 x(2)=x(1)+C4Mleft y(2)=y(1)+C4Mtop width(0) = width(2) +C4Mleft+C4Mright height(0)=height(2) +C4Mtop+C4Mbottom
C4-2	$[\text{c4 root}]_0$:= $[\text{c4 row}]_1$ $[\text{c4 root}]_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) y(2)=y(1) + height(1) + C4Sv width(0) = max(width(1),width(2)) height(0)= height(1)+height(2) +C4Sv
C4-3	$[\text{c4 root}]_0$:= $[\text{c4 row}]_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
C4-4	$[\text{c4 row}]_0$:= $[\text{c4 column}]_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
C4-5	$[\text{c4 column}]_0$:= $[\text{c4 scalar}]_1$ $[\text{c4 column}]_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) +width(1)+C4Sh y(2)=y(1) width(0) = width(1)+width(2)+C4Sh height(0)= max(height(1),height(2))
C4-6	$[\text{c4 column}]_0$:= $[\text{c4 scalar}]_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
C4-7	$[\text{c4 scalar}]_0$:= $[\text{Term. \& Comv.}]_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_termconv height(0)= HEIGHT_termconv
C4-8	$[\text{c4 scalar}]_0$:= $[\text{Procedure Struct.}]_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_procstruct height(0)= HEIGHT_procstruct
C4-9	$[\text{c4 scalar}]_0$:= $[\text{Definition Prc.}]_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_defprocedure height(0)= HEIGHT_defprocedure

No.	Production	Semantic rule
C5-0	$[\text{c body}]_0$:= $[\text{c5}]_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
C5-1	$[\text{c5}]_0$:= C5_1 $[\text{c5 root}]_2$	x(1)=0 y(1)=0 x(2)=x(1)+C5Mleft y(2)=y(1)+C5Mtop width(0) = width(2) +C5Mleft+C5Mright height(0)=height(2) +C5Mtop+C5Mbottom
C5-2	$[\text{c5 root}]_0$:= $[\text{c5 row}]_1$ $[\text{c5 root}]_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) y(2)=y(1) + height(1) + C5Sv width(0) = max(width(1),width(2)) height(0)= height(1)+height(2) +C5Sv
C5-3	$[\text{c5 root}]_0$:= $[\text{c5 row}]_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
C5-4	$[\text{c5 row}]_0$:= $[\text{c5 column}]_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
C5-5	$[\text{c5 column}]_0$:= $[\text{c5 scalar}]_1$ $[\text{c5 column}]_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) +width(1)+C5Sh y(2)=y(1) width(0) = width(1)+width(2)+C5Sh height(0)= max(height(1),height(2))
C5-6	$[\text{c5 column}]_0$:= $[\text{c5 scalar}]_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
C5-7	$[\text{c5 scalar}]_0$:= $[\text{D. of Procedure}]_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_dprocedure height(0)= HEIGHT_dprocedure

No.	Production	Semantic rule
C6-0	$[\text{c body}]_0$:= $[\text{c6}]_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
C6-1	$[\text{c6}]_0$:= C6_1 $[\text{c6 root}]_2$	x(1)=0 y(1)=0 x(2)=x(1)+C6Mleft y(2)=y(1)+C6Mtop width(0) = width(2) +C6Mleft+C6Mright height(0)=height(2) +C6Mtop+C6Mbottom
C6-2	$[\text{c6 root}]_0$:= $[\text{c6 row}]_1$ $[\text{c6 root}]_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) y(2)=y(1) + height(1) + C6Sv width(0) = max(width(1),width(2)) height(0)= height(1)+height(2) +C6Sv
C6-3	$[\text{c6 root}]_0$:= $[\text{c6 row}]_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
C6-4	$[\text{c6 row}]_0$:= $[\text{c6 column}]_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
C6-5	$[\text{c6 column}]_0$:= $[\text{c6 scalar}]_1$ $[\text{c6 column}]_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) +width(1)+C6Sh y(2)=y(1) width(0) = width(1)+width(2)+C6Sh height(0)= max(height(1),height(2))
C6-6	$[\text{c6 column}]_0$:= $[\text{c6 scalar}]_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
C6-7	$[\text{c6 scalar}]_0$:= $[\text{Ex. of Precedure}]_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_exampleprocedure height(0)= HEIGHT_exampleprocedure

No.	Production	Semantic rule
D1-0	$[\text{body}]_0$:= $[\text{d1}]_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
D1-1	$[\text{d1}]_0$:= D1_1 $[\text{d1 root}]_2$	x(1)=0 y(1)=0 x(2)=x(1)+D1Mleft y(2)=y(1)+D1Mtop width(0) = width(2) +D1Mleft+D1Mright height(0)=height(2) +D1Mtop+D1Mbottom
D1-2	$[\text{d1 root}]_0$:= $[\text{d1 row}]_1$ $[\text{d1 root}]_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) y(2)=y(1) + height(1) + D1Sv width(0) = max(width(1),width(2)) height(0)= height(1)+height(2) +D1Sv
D1-3	$[\text{d1 root}]_0$:= $[\text{d1 row}]_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
D1-4	$[\text{d1 row}]_0$:= $[\text{d1 column}]_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
D1-5	$[\text{d1 column}]_0$:= $[\text{d1 scalar}]_1$ $[\text{d1 column}]_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) +width(1)+D1Sh y(2)=y(1) width(0) = width(1)+width(2)+D1Sh height(0)= max(height(1),height(2))
D1-6	$[\text{d1 column}]_0$:= $[\text{d1 scalar}]_1$	x(1)=x(0) y(1)=y(0) width(0) = width(1) height(0)= height(1)
D1-7	$[\text{d1 scalar}]_0$:= $[\text{D. of Prog. Struct.}]_1$	x(1)=x(0) y(1)=y(0) width(0) = WIDTH_dprogramstructure height(0)= HEIGHT_dprogramstructure

No.	Production	Semantic rule	
D2-0	[body]$_0$:= [d2]$_1$	x(1)=x(0) y(1)=y(0)	width(0) = width(1) height(0)= height(1)
D2-1	[d2]$_0$:= D2$_1$ [d2 root]$_2$	x(1)=0 y(1)=0 x(2)=x(1)+D2Mleft y(2)=y(1)+D2Mtop	width(0) = width(2) +D2Mleft+D2Mright height(0)=height(2) +D2Mtop+D2Mbottom
D2-2	[d2 root]$_0$:= [d2 row]$_1$ [d2 root]$_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) y(2)=y(1) + height(1) + D2Sv	width(0) = max(width(1),width(2)) height(0)= height(1)+height(2) +D2Sv
D2-3	[d2 root]$_0$:= [d2 row]$_1$	x(1)=x(0) y(1)=y(0)	width(0) = width(1) height(0)= height(1)
D2-4	[d2 row]$_0$:= [d2 column]$_1$	x(1)=x(0) y(1)=y(0)	width(0) = width(1) height(0)= height(1)
D2-5	[d2 column]$_0$:= [d2 scalar]$_1$ [d2 column]$_2$	x(1)=x(0) y(1)=y(0) x(2)=x(1) +width(1)+D2Sh y(2)=y(1)	width(0) = width(1)+width(2)+D2Sh height(0)= max(height(1),height(2))
D2-6	[d2 column]$_0$:= [d2 scalar]$_1$	x(1)=x(0) y(1)=y(0)	width(0) = width(1) height(0)= height(1)
D2-7	[d2 scalar]$_0$:= [Module Name]$_1$	x(1)=x(0) y(1)=y(0)	width(0) = WIDTH_modulename height(0)= HEIGHT_modulename
D2-8	[d2 scalar]$_0$:= [Module Version]$_1$	x(1)=x(0) y(1)=y(0)	width(0) = WIDTH_moduleversion height(0)= HEIGHT_moduleversion
D2-9	[d2 scalar]$_0$:= [Module Author]$_1$	x(1)=x(0) y(1)=y(0)	width(0) = WIDTH_moduleauthor height(0)= HEIGHT_moduleauthor
D2-10	[d2 scalar]$_0$:= [Module Release]$_1$	x(1)=x(0) y(1)=y(0)	width(0) = WIDTH_modulerelease height(0)= HEIGHT_modulerelease
D2-11	[d2 scalar]$_0$:= [Varients]$_1$	x(1)=x(0) y(1)=y(0)	width(0) = WIDTH_varient height(0)= HEIGHT_varient
D2-12	[d2 scalar]$_0$:= [Key Words]$_1$	x(1)=x(0) y(1)=y(0)	width(0) = WIDTH_keyword height(0)= HEIGHT_keyword
D2-13	[d2 scalar]$_0$:= [Size]$_1$	x(1)=x(0) y(1)=y(0)	width(0) = WIDTH_size height(0)= HEIGHT_size
D2-14	[d2 scalar]$_0$:= [Media]$_1$	x(1)=x(0) y(1)=y(0)	width(0) = WIDTH_media height(0)= HEIGHT_media
D2-15	[d2 scalar]$_0$:= [Objective]$_1$	x(1)=x(0) y(1)=y(0)	width(0) = WIDTH_objective height(0)= HEIGHT_objective
D2-16	[d2 scalar]$_0$:= [Method]$_1$	x(1)=x(0) y(1)=y(0)	width(0) = WIDTH_method height(0)= HEIGHT_method
D2-17	[d2 scalar]$_0$:= [References]$_1$	x(1)=x(0) y(1)=y(0)	width(0) = WIDTH_reference height(0)= HEIGHT_reference

No.	Production	Semantic rule	
D2-18	[d2 scalar]$_0$:= [Language]$_1$	x(1)=x(0) y(1)=y(0)	width(0) = WIDTH_language height(0)= HEIGHT_language
D2-19	[d2 scalar]$_0$:= [Software Req.]$_1$	x(1)=x(0) y(1)=y(0)	width(0) = WIDTH_softwarereq height(0)= HEIGHT_softwarereq
D2-20	[d2 scalar]$_0$:= [Result D. Descript.]$_1$	x(1)=x(0) y(1)=y(0)	width(0) = WIDTH_resultddescription height(0)= HEIGHT_resultddescription
D2-21	[d2 scalar]$_0$:= [Invoking Specif.]$_1$	x(1)=x(0) y(1)=y(0)	width(0) = WIDTH_invokingspecif height(0)= HEIGHT_invokingspecif
D2-22	[d2 scalar]$_0$:= [Example Invoking]$_1$	x(1)=x(0) y(1)=y(0)	width(0) = WIDTH_exofinvoking height(0)= HEIGHT_exofinvoking
D2-23	[d2 scalar]$_0$:= [Inter Consist.]$_1$	x(1)=x(0) y(1)=y(0)	width(0) = WIDTH_interconsistency height(0)= HEIGHT_interconsistency
D2-24	[d2 scalar]$_0$:= [Data Sharing Spe.]$_1$	x(1)=x(0) y(1)=y(0)	width(0) = WIDTH_datasharing height(0)= HEIGHT_datasharing

Right / Left	A1 Terminal Sym.			[a1 scalar]			[a1 column]			[a1 row]			[a1 root]			A1			[a1]		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
A1 Terminal Sym.		<·>	<·>		<·>	<·>		<·>	⋗		<·>			⋗		⋗					
[a1 scalar]		<·>	⋖		<·>	⋖		<·>	≐		<·>			⋗		⋗					
[a1 column]		<·>			<·>			<·>			<·>			⋗		⋗					
[a1 row]		⋖			⋖			⋖			⋖			≐		⋗					
[a1 root]																≐					
[a1]																					
A1																					

Right / Left	A2 Terminal Sym.			[a2 scalar]			[a2 column]			[a2 row]			[a2 root]			A2			[a2]		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
A2 Terminal Sym.		<·>	<·>		<·>	<·>		<·>	⋗		<·>			⋗		⋗					
[a2 scalar]		<·>	⋖		<·>	⋖		<·>	≐		<·>			⋗		⋗					
[a2 column]		<·>			<·>			<·>			<·>			⋗		⋗					
[a2 row]		⋖			⋖			⋖			⋖			≐		⋗					
[a2 root]																≐					
[a2]																					
A2																					

Right / Left	a3 Terminal Sym.			[a3 scalar]			[a3 column]			[a3 row]			[a3 root]			A3			[a3]		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
A3 Terminal Sym.		<·>	<·>		<·>	<·>		<·>	⋗		<·>			⋗		⋗					
[a3 scalar]		<·>	⋖		<·>	⋖		<·>	≐		<·>			⋗		⋗					
[a3 column]		<·>			<·>			<·>			<·>			⋗		⋗					
[a3 row]		⋖			⋖			⋖			⋖			≐		⋗					
[a3 root]																≐					
[a3]																					
A3																					

Right / Left	A4 Terminal Sym.			[a4 scalar]			[a4 column]			[a4 row]			[a4 root]			A4			[a4]		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
A4 Terminal Sym.		<·>	<·>		<·>	<·>		<·>	⋗		<·>			⋗		⋗					
[a4 scalar]		<·>	⋖		<·>	⋖		<·>	≐		<·>			⋗		⋗					
[a4 column]		<·>			<·>			<·>			<·>			⋗		⋗					
[a4 row]		⋖			⋖			⋖			⋖			≐		⋗					
[a4 root]																≐					
[a4]																					
A4																					

Right / Left	A5 Terminal Sym.			[a5 scalar]			[a5 column]			[a5 row]			[a5 root]			A5			[a5]		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
A5 Terminal Sym.		<·>	<·>		<·>	<·>		<·>	⋗		<·>			⋗		⋗					
[a5 scalar]		<·>	⋖		<·>	⋖		<·>	≐		<·>			⋗		⋗					
[a5 column]		<·>			<·>			<·>			<·>			⋗		⋗					
[a5 row]		⋖			⋖			⋖			⋖			≐		⋗					
[a5 root]																≐					
[a5]																					
A5																					

Right / Left	A6 Terminal Sym.			[a6 scalar]			[a6 column]			[a6 row]			[a6 root]			A6			[a6]		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
A6 Terminal Sym.		<·>	<·>		<·>	<·>		<·>	⋗		<·>			⋗		⋗					
[a6 scalar]		<·>	⋖		<·>	⋖		<·>	≐		<·>			⋗		⋗					
[a6 column]		<·>			<·>			<·>			<·>			⋗		⋗					
[a6 row]		⋖			⋖			⋖			⋖			≐		⋗					
[a6 root]																≐					
[a6]																					
A6																					

Right / Left	D. of Data Doc.			[b1 scalar]			[b1 column]			[b1 row]			[b1 root]			B1			[b1]		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
D. of Data Doc.		<·>	<·>		<·>	<·>		<·>	⋗		<·>			⋗		⋗					
[b1 scalar]		<·>	⋖		<·>	⋖		<·>	≐		<·>			⋗		⋗					
[b1 column]		<·>			<·>			<·>			<·>			⋗		⋗					
[b1 row]		⋖			⋖			⋖			⋖			≐		⋗					
[b1root]																≐					
[b1]																					
B1																					

Right \ Left	B2 Terminal Sym.			[b2 scalar]			[b2 column]			[b2 row]			[b2 root]			B2			[b2]		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
B2 Terminal Sym.		<·>	<·>		<·>	<·>		<·>	⋗		<·>			⋗		⋗					
[b2 scalar]		<·>	⋖		<·>	⋖		<·>	≐		<·>			⋗		⋗					
[b2 column]		<·>			<·>			<·>			<·>			⋗		⋗					
[b2 row]		⋖			⋖			⋖			⋖			≐		⋗					
[b2 root]																≐					
[b2]																					
B2																					

Right \ Left	B3 Terminal Sym.			[b3 scalar]			[b3 column]			[b3 row]			[b3 root]			B3			[b3]		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
B3 Terminal Sym.		<·>	<·>		<·>	<·>		<·>	⋗		<·>			⋗		⋗					
[b3 scalar]		<·>	⋖		<·>	⋖		<·>	≐		<·>			⋗		⋗					
[b3 column]		<·>			<·>			<·>			<·>			⋗		⋗					
[b3 row]		⋖			⋖			⋖			⋖			≐		⋗					
[b3 root]																≐					
[b3]																					
B3																					

Right \ Left	D. of Procedure			[c1 scalar]			[c1 column]			[c1 row]			[c1 root]			C1			[c1]		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
D. of Procedure		<·>	<·>		<·>	<·>		<·>	⋗		<·>			⋗		⋗					
[c1 scalar]		<·>	⋖		<·>	⋖		<·>	≐		<·>			⋗		⋗					
[c1 column]		<·>			<·>			<·>			<·>			⋗		⋗					
[c1 row]		⋖			⋖			⋖			⋖			≐		⋗					
[c1root]																≐					
[c1]																					
C1																					

Left \ Right	B			[bc body]			[b body]			[b2]			[b3]			BC			B2			B3		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
B																								
[bc body]	≐							≐			⋖			⋖						⋖			⋖	
[b body]																								
[b2]																								
[b3]																								
BC	⋗							⋗			⋖⋗			⋖⋗						⋖⋗			⋖⋗	
B2																								
B3																								

Left \ Right	Tecnical Id.			Appli. Orient. Id.			[bc scalar]			[bc column]			[bc row]			[bc root]			BC		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
Tecnical Id.		⋖⋗	⋖⋗		⋖⋗	⋖⋗		⋖⋗	⋖⋗		⋖⋗	⋗		⋖⋗			⋗		⋗		
Appli. Orient. Id.		⋖⋗	⋖⋗		⋖⋗	⋖⋗		⋖⋗	⋖⋗		⋖⋗	⋗		⋖⋗			⋗		⋗		
[bc scalar]		⋖⋗	⋖		⋖⋗	⋖		⋖⋗	⋖		⋖⋗	≐		⋖⋗			⋗		⋗		
[bc column]		⋖⋗			⋖⋗			⋖⋗			⋖⋗			⋖⋗			⋗		⋗		
[bc row]		⋖			⋖			⋖			⋖			⋖			≐		⋗		
[bc root]																			≐		
BC																					

Right / Left	C2 Terminal Sym.			[c2 scalar]			[c2 column]			[c2 row]			[c2 root]			C2			[c2]		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
C2 Terminal Sym.		<·>	<·>		<·>	<·>		<·>	⋗		<·>			⋗		⋗					
[c2 scalar]		<·>	⋖		<·>	⋖		<·>	≐		<·>			⋗		⋗					
[c2 column]		<·>			<·>			<·>			<·>			⋗		⋗					
[c2 row]		⋖			⋖			⋖			⋖			≐		⋗					
[c2 root]																≐					
[c2]																					
C2																					

Right / Left	C3Terminal Sym.			[c3 scalar]			[c3 column]			[c3 row]			[c3 root]			C3			[c3]		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
C3 Terminal Sym.		<·>	<·>		<·>	<·>		<·>	⋗		<·>			⋗		⋗					
[c3 scalar]		<·>	⋖		<·>	⋖		<·>	≐		<·>			⋗		⋗					
[c3 column]		<·>			<·>			<·>			<·>			⋗		⋗					
[c3 row]		⋖			⋖			⋖			⋖			≐		⋗					
[c3 root]																≐					
[c3]																					
C3																					

Right / Left	C4 Terminal Sym.			[c4 scalar]			[c4 column]			[c4 row]			[c4 root]			C4			[c4]		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
C4 Terminal Sym.		<·>	<·>		<·>	<·>		<·>	⋗		<·>			⋗		⋗					
[c4 scalar]		<·>	⋖		<·>	⋖		<·>	≐		<·>			⋗		⋗					
[c4 column]		<·>			<·>			<·>			<·>			⋗		⋗					
[c4 row]		⋖			⋖			⋖			⋖			≐		⋗					
[c4 root]																≐					
[c4]																					
C4																					

Right / Left	C5 Terminal Sym.			[c5 scalar]			[c5 column]			[c5 row]			[c5 root]			C5			[c5]		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
C5 Terminal Sym.		<·>	<·>		<·>	<·>		<·>	⋗		<·>			⋗		⋗					
[c5 scalar]		<·>	⋖		<·>	⋖		<·>	≐		<·>			⋗		⋗					
[c5 column]		<·>			<·>			<·>			<·>			⋗		⋗					
[c5 row]		⋖			⋖			⋖			⋖			≐		⋗					
[c5 root]																≐					
[c5]																					
C5																					

Right / Left	C6 Terminal Sym.			[c6 scalar]			[c6 column]			[c6 row]			[c6 root]			C6			[c6]		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
C6 Terminal Sym.		<·>	<·>		<·>	<·>		<·>	⋗		<·>			⋗		⋗					
[c6 scalar]		<·>	⋖		<·>	⋖		<·>	≐		<·>			⋗		⋗					
[c6 column]		<·>			<·>			<·>			<·>			⋗		⋗					
[c6 row]		⋖			⋖			⋖			⋖			≐		⋗					
[c6 root]																≐					
[c6]																					
C6																					

Right / Left	D1 Terminal Sym.			[d1 scalar]			[d1 column]			[d1 row]			[d1 root]			D1			[d1]		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
D1 Terminal Sym.		<·>	<·>		<·>	<·>		<·>	⋗		<·>			⋗		⋗					
[d1 scalar]		<·>	⋖		<·>	⋖		<·>	≐		<·>			⋗		⋗					
[d1 column]		<·>			<·>			<·>			<·>			⋗		⋗					
[d1 row]		⋖			⋖			⋖			⋖			≐		⋗					
[d1 root]																≐					
[d1]																					
D1																					

Right / Left	D2 Terminal Sym.			[d2 scalar]			[d2 column]			[d2 row]			[d2 root]			D2			[d2]		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
D2Terminal Sym.		<·>	<·>		<·>	<·>		<·>	⋗		<·>			⋗		⋗					
[d2 scalar]		<·>	⋖		<·>	⋖		<·>	≐		<·>			⋗		⋗					
[d2 column]		<·>			<·>			<·>			<·>			⋗		⋗					
[d2 row]		⋖			⋖			⋖			⋖			≐		⋗					
[d2 root]																≐					
[d2]																					
D2																					

Right / Left	C			[cc body]			[c body]			[c2]			[c3]			[c4]			[c5]			[c6]		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
C																								
[cc body]	≐							≐			⋖			⋖			⋖			⋖			⋖	
[c body]																								
[c2]																								
[c3]																								
[c4]																								
[c5]																								
[c6]																								
CC	⋗							⋗			<·>			<·>			<·>			<·>			<·>	
C2																								
C3																								
C4																								
C5																								
C6																								

Right / Left	CC			C2			C3			C4			C5			C6		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
C																		
[cc body]					⋖			⋖			⋖			⋖			⋖	
[c body]																		
[c2]																		
[c3]																		
[c4]																		
[c5]																		
[c6]																		
CC					<·>			<·>			<·>			<·>			<·>	
C2																		
C3																		
C4																		
C5																		
C6																		

Right / Left	Variants			Procedure name			[cc scalar]			[cc column]			[cc row]			[cc root]			CC		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
Variants		<·>	<·>		<·>	<·>		<·>	<·>		<·>	⋗		<·>			⋗		⋗		
Procedure name		<·>	<·>		<·>	<·>		<·>	<·>		<·>	⋗		<·>			⋗		⋗		
[cc scalar]		<·>	⋖		<·>	⋖		<·>	⋖		<·>	≐		<·>			⋗		⋗		
[cc column]		<·>			<·>			<·>			<·>			<·>			⋗		⋗		
[cc row]		⋖			⋖			⋖			⋖			⋖			≐		⋗		
[cc root]																			≐		
CC																					

Right / Left	Head Terminal Sym.			[head scalar]			[head column]			[head row]			[head root]			Head			[head]		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
Head Terminal Sym.		<·>	<·>		<·>	<·>		<·>	⋗		<·>			⋗		⋗					
[head scalar]		<·>	⋖		<·>	⋖		<·>	≐		<·>			⋗		⋗					
[head column]		<·>			<·>			<·>			<·>			⋗		⋗					
[head row]		⋖			⋖			⋖			⋖			≐		⋗					
[head root]																≐					
[head]																					
Head																					

Right / Left				[innerstruct]			[head]			HEAD			[body]			[a1]			[a2]		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
[innerstruct]	≐																				
[head]	⋗													≐			⋖			⋖	
HEAD	⋗													⋗			<·>			<·>	
[body]																					
[a1]																					
[a2]																					
[a3]																					
[a4]																					
[a5]																					
[a6]																					
[b]																					
[b1]																					
[c]																					
[c1]																					
[d1]																					
[d2]																					
A1																					
A2																					
A3																					
A4																					
A5																					
A6																					
B																					
B1																					
C																					
C1																					
D1																					
D2																					

Right / Left	[a3]			[a4]			[a5]			[a6]			[b]			[b1]			[c]		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
[innerstruct]																					
[head]		<.			<.			<.			<.			<.			<.			<.	
HEAD		<.>			<.>			<.>			<.>			<.>			<.>			<.>	
[body]																					
[a1]																					
[a2]																					
[a3]																					
[a4]																					
[a5]																					
[a6]																					
[b]																					
[b1]																					
[c]																					
[c1]																					
[d1]																					
[d2]																					
A1																					
A2																					
A3																					
A4																					
A5																					
A6																					
B																					
B1																					
C																					
C1																					
D1																					
D2																					

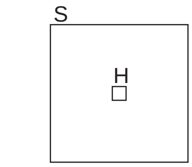
Right / Left	[c1]			[d1]			[d2]			A1			A2			A3			A4		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
[innerstruct]																					
[head]		<.			<.			<.			<.			<.			<.			<.	
HEAD		<.>			<.>			<.>			<.>			<.>			<.>			<.>	
[body]																					
[a1]																					
[a2]																					
[a3]																					
[a4]																					
[a5]																					
[a6]																					
[b]																					
[b1]																					
[c]																					
[c1]																					
[d1]																					
[d2]																					
A1																					
A2																					
A3																					
A4																					
A5																					
A6																					
B																					
B1																					
C																					
C1																					
D1																					
D2																					

Right / Left	A5			A6			B			B1			C			C1			D1			D2		
	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf	in	ov	lf
[innerstruct]																								
[head]		<.			<.			<.			<.			<.			<.			<.			<.	
HEAD		<.>			<.>			<.>			<.>			<.>			<.>			<.>			<.>	
[body]																								
[a1]																								
[a2]																								
[a3]																								
[a4]																								
[a5]																								
[a6]																								
[b]																								
[b1]																								
[c]																								
[c1]																								
[d1]																								
[d2]																								
A1																								
A2																								
A3																								
A4																								
A5																								
A6																								
B																								
B1																								
C																								
C1																								
D1																								
D2																								

Appendix B

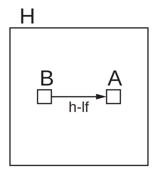
An Attribute Graph Grammar for Tessellation Forms

Productions and Semantic Rules for Tessellation Forms (Horizontal Derivation 1)



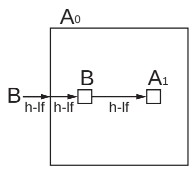
x(H) = 0
y(H) = 0

width(S) = width(H)
height(S) = height(H)



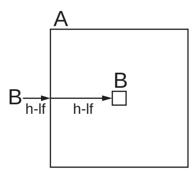
x(B) = x(A)
y(B) = y(A)
x(A) = x(H)+width(B)
y(A) = y(H)

width(H) = width(B)+width(A)
height(H) = max(height(B), height(A))



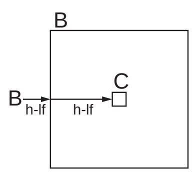
$x(B) = x(A_0)$
$y(B) = y(A_0)$
$x(A_1) = x(A_0)+width(B)$
$y(A_1) = y(A_0)$

$width(A_0) = width(B)+width(A_1)$
$height(A_0) = max(height(B), height(A_1))$



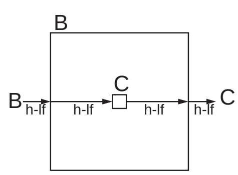
x(B) = x(A)
y(B) = y(A)

width(A) = width(B)
height(A) = height(B)



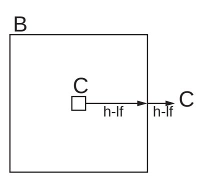
x(C) = x(B)
y(C) = y(B)

width(B) = width(C)
height(B) = height(C)



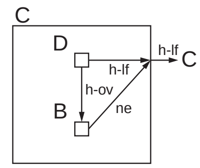
x(C) = x(B)
y(C) = y(B)

width(B) = width(C)
height(B) = height(C)



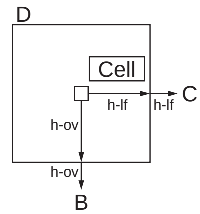
x(C) = x(B)
y(C) = y(B)

width(B) = width(C)
height(B) = height(C)



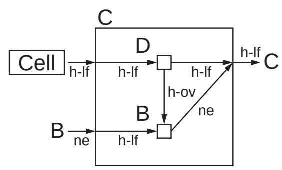
x(D) = x(C)
y(D) = y(C)
x(B) = x(C)
y(B) = y(C)+height(D)

width(C) = max(width(D) ,width(B))
height(C) = height(D)+height(B)



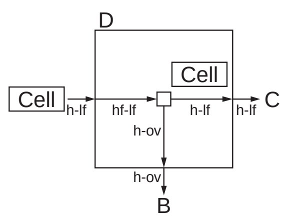
x(Cell) = x(D)
y(Cell) = y(D)

width(D) = WIDTH_cell
height(D) = HEIGHT_cell



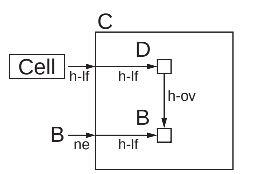
x(D) = x(C)
y(D) = y(C)
x(B) = x(C)
y(B) = y(C)+height(D)

width(C)
= max(width(D) ,width(B))
height(C) = height(D)+height(B)



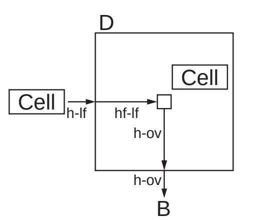
x(Cell) = x(D)
y(Cell) = y(D)

width(D) = WIDTH_cell
height(D) = HEIGHT_cell



x(D) = x(C)
y(D) = y(C)
x(B) = x(C)
y(B) = y(C)+height(D)

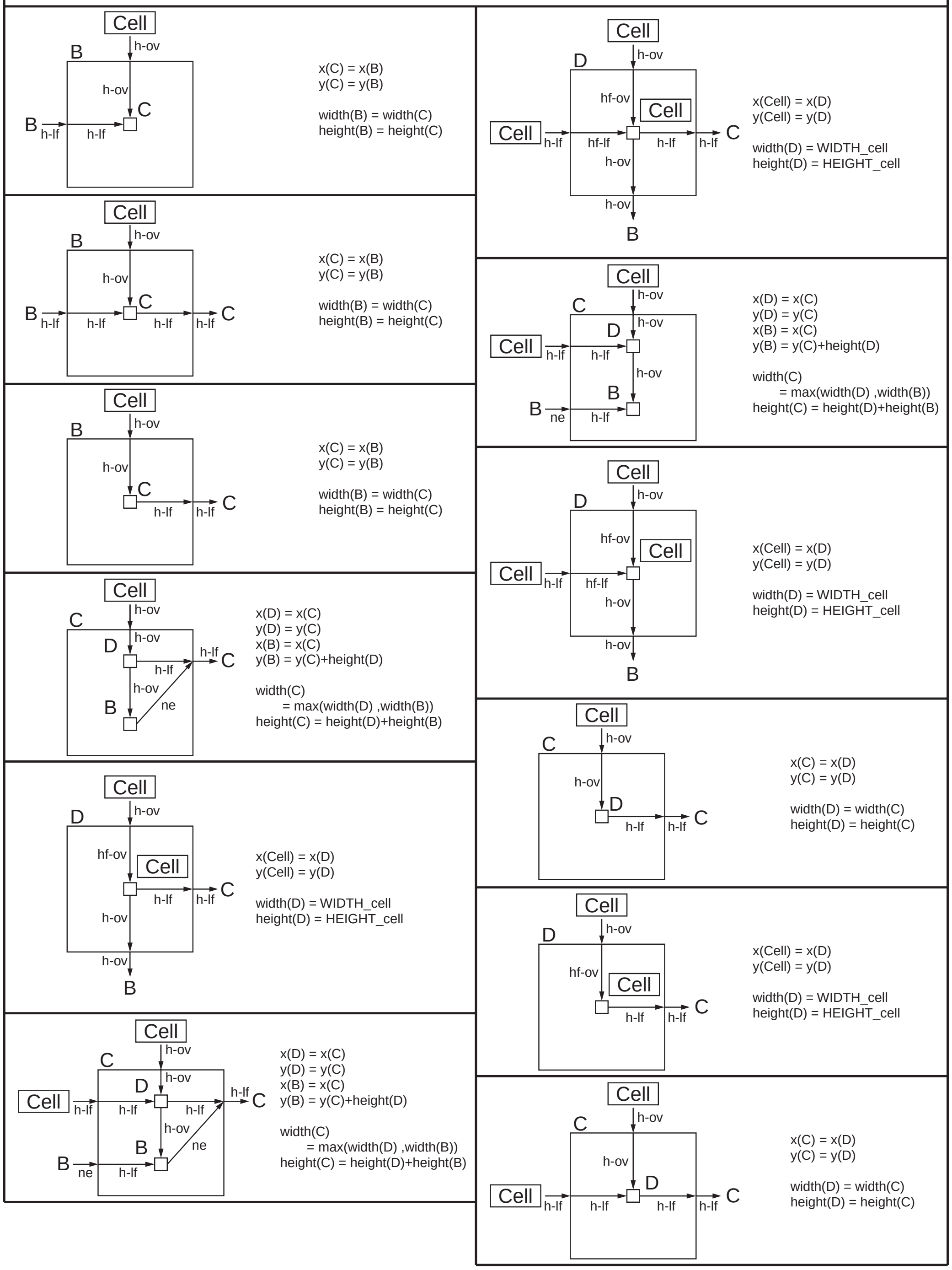
width(C)
= max(width(D) ,width(B))
height(C) = height(D)+height(B)



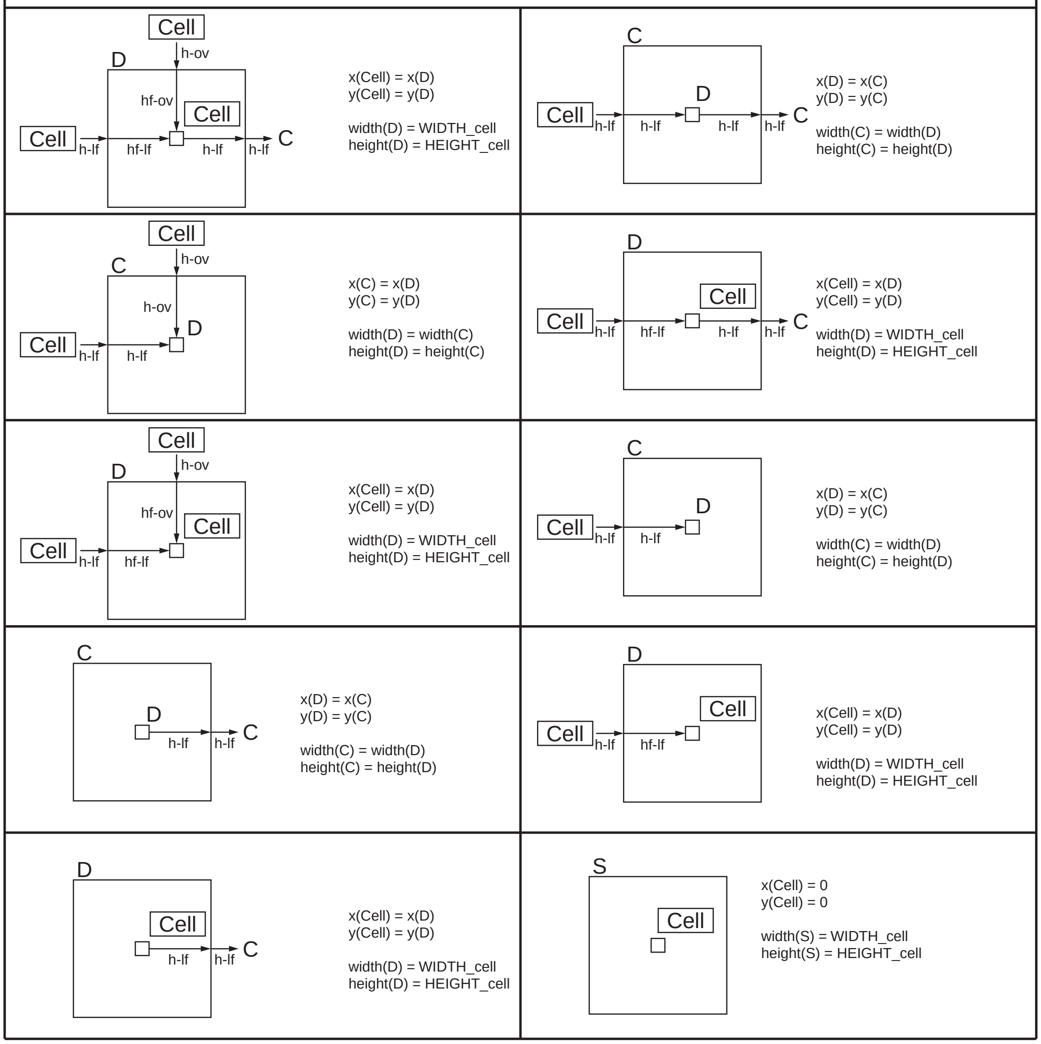
x(Cell) = x(D)
y(Cell) = y(D)

width(D) = WIDTH_cell
height(D) = HEIGHT_cell

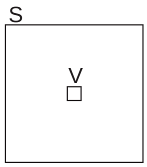
Productions and Semantic Rules for Tessellation Forms (Horizontal Derivation 2)



Productions and Semantic Rules for Tessellation Forms (Horizontal Derivation 3)

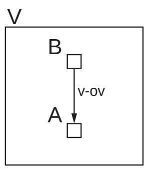


Productions and Semantic Rules for Tessellation Forms (Vertical Derivation 1)



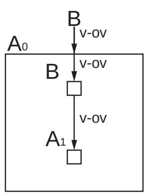
x(V) = 0
y(V) = 0

width(S) = width(V)
height(S) = height(V)



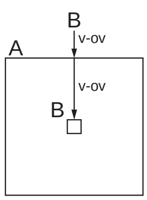
x(B) = x(V)
y(B) = y(V)
x(A) = x(V)
y(A) = y(V) + height(B)

width(V) = max(width(B), width(A))
height(V) = height(B) + height(A)



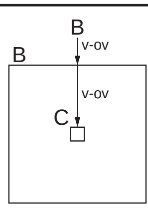
$x(B) = x(A_0)$
$y(B) = y(A_0)$
$x(A_1) = x(A_0)$
$y(A_1) = y(A_0) + height(B)$

$width(A_0) = max(width(B), width(A_1))$
$height(A_0) = height(B) + height(A_1)$



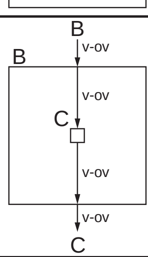
x(B) = x(A)
y(B) = y(A)

width(A) = width(B)
height(A) = height(B)



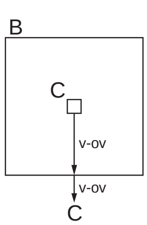
x(C) = x(B)
y(C) = y(B)

width(B) = width(A)
height(B) = height(A)



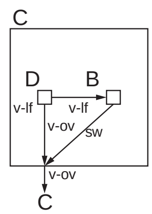
x(C) = x(B)
y(C) = y(B)

width(B) = width(A)
height(B) = height(A)



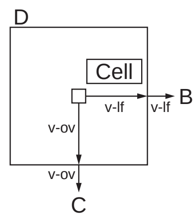
x(C) = x(B)
y(C) = y(B)

width(B) = width(A)
height(B) = height(A)



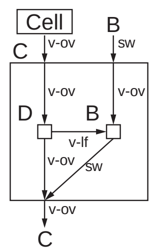
x(D) = x(C)
y(D) = y(C)
x(B) = x(C) + width(D)
y(B) = y(C)

width(C) = width(D) + width(B)
height(C) = max(height(D), height(B))



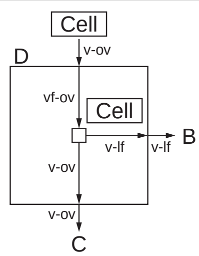
x(Cell) = x(D)
y(Cell) = y(D)

width(D) = WIDTH_cell
height(D) = HEIGHT_cell



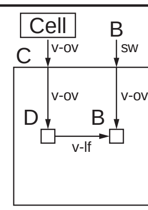
x(D) = x(C)
y(D) = y(C)
x(B) = x(C) + width(D)
y(B) = y(C)

width(C) = width(D) + width(B)
height(C) = max(height(D), height(B))



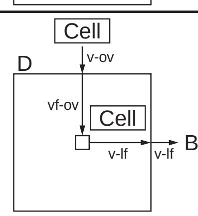
x(Cell) = x(D)
y(Cell) = y(D)

width(D) = WIDTH_cell
height(D) = HEIGHT_cell



x(D) = x(C)
y(D) = y(C)
x(B) = x(C) + width(D)
y(B) = y(C)

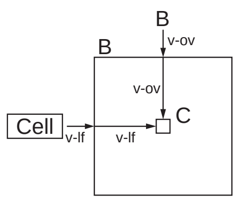
width(C) = width(D) + width(B)
height(C) = max(height(D), height(B))



x(Cell) = x(D)
y(Cell) = y(D)

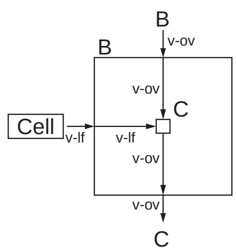
width(D) = WIDTH_cell
height(D) = HEIGHT_cell

Productions and Semantic Rules for Tessellation Forms (Vertical Derivation 2)



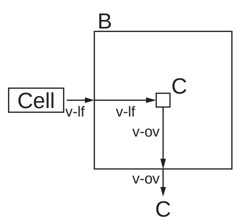
x(C) = x(B)
y(C) = y(B)

width(B) = width(C)
height(B) = height(C)



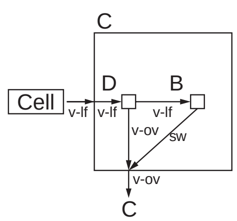
x(C) = x(B)
y(C) = y(B)

width(B) = width(C)
height(B) = height(C)



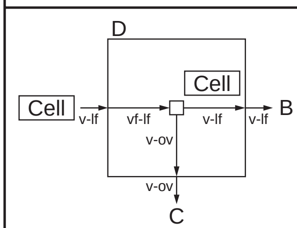
x(C) = x(B)
y(C) = y(B)

width(B) = width(C)
height(B) = height(C)



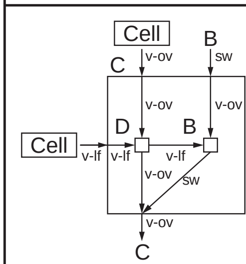
x(D) = x(C)
y(D) = y(C)
x(B) = x(C) + width(D)
y(B) = y(C)

width(C) = width(D) + width(B)
height(C) = max(height(D), height(B))



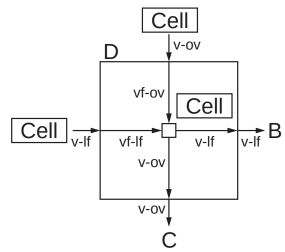
x(Cell) = x(D)
y(Cell) = y(D)

width(D) = WIDTH_cell
height(D) = HEIGHT_cell



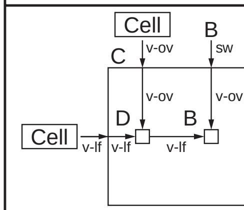
x(D) = x(C)
y(D) = y(C)
x(B) = x(C) + width(D)
y(B) = y(C)

width(C) = width(D) + width(B)
height(C) = max(height(D), height(B))



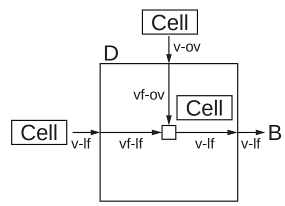
x(Cell) = x(D)
y(Cell) = y(D)

width(D) = WIDTH_cell
height(D) = HEIGHT_cell



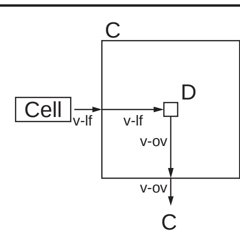
x(D) = x(C)
y(D) = y(C)
x(B) = x(C) + width(D)
y(B) = y(C)

width(C) = width(D) + width(B)
height(C) = max(height(D), height(B))



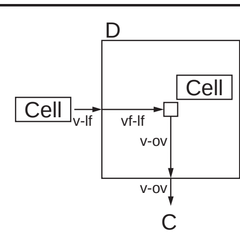
x(Cell) = x(D)
y(Cell) = y(D)

width(D) = WIDTH_cell
height(D) = HEIGHT_cell



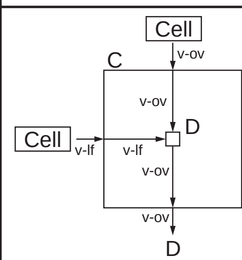
x(D) = x(C)
y(D) = y(C)

width(C) = width(D)
height(C) = height(D)



x(Cell) = x(D)
y(Cell) = y(D)

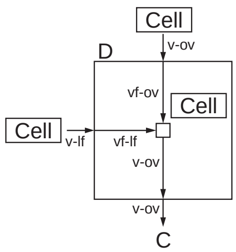
width(D) = WIDTH_cell
height(D) = HEIGHT_cell



x(D) = x(C)
y(D) = y(C)

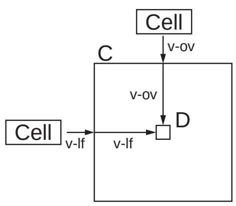
width(C) = width(D)
height(C) = height(D)

Productions and Semantic Rules for Tessellation Forms (Vertical Derivation 3)



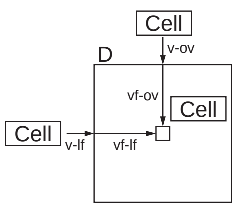
x(Cell) = x(D)
y(Cell) = y(D)

width(D) = WIDTH_cell
height(D) = HEIGHT_cell



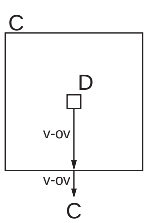
x(D) = x(C)
y(D) = y(C)

width(C) = width(D)
height(C) = height(D)



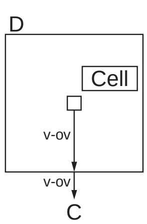
x(Cell) = x(D)
y(Cell) = y(D)

width(D) = WIDTH_cell
height(D) = HEIGHT_cell



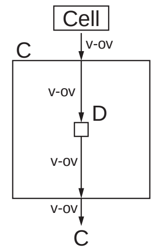
x(D) = x(C)
y(D) = y(C)

width(C) = width(D)
height(C) = height(D)



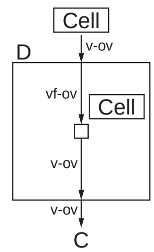
x(Cell) = x(D)
y(Cell) = y(D)

width(D) = WIDTH_cell
height(D) = HEIGHT_cell



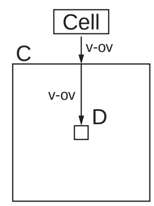
x(D) = x(C)
y(D) = y(C)

width(C) = width(D)
height(C) = height(D)



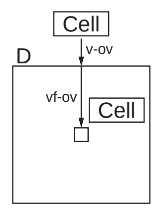
x(Cell) = x(D)
y(Cell) = y(D)

width(D) = WIDTH_cell
height(D) = HEIGHT_cell



x(D) = x(C)
y(D) = y(C)

width(C) = width(D)
height(C) = height(D)



x(Cell) = x(D)
y(Cell) = y(D)

width(D) = WIDTH_cell
height(D) = HEIGHT_cell