Syntactic Editing of Tabular Forms by Attribute edNCE Graph Grammars

Kiyonobu TOMIYAMA

6199M11

Target

program name : Hanoi_main	**A** **General document**
subtitle : hanoi	
library code : cs - 2000 - 01	**version :** 1.0
author : Kiyonobu Tomiyama	**original release :** 1999/12/22
approve :	**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)
```

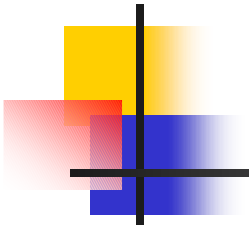
2. Example of Output

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

Name	Type	Size	G/L
x	int	2	G
y	float	4	L

Tessellation Tabular Form

Program Specification Hiform



The contents

1. Introduction

2. Preliminaries

   Tabular Form, Graph Grammars

   Composition of Production Copies, Confluence Property

3. Editing of Nested Tabular Form

3.1 Production Instance 3.2 Syntactic Insertion

3.3 Syntactic Addition 3.4 Syntactic Deletion of Item

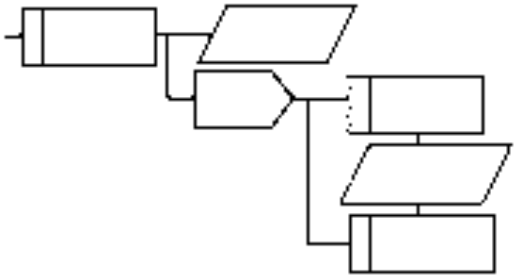
4. Example：Insertion Process

5. Editing of Tessellation Tabular Forms

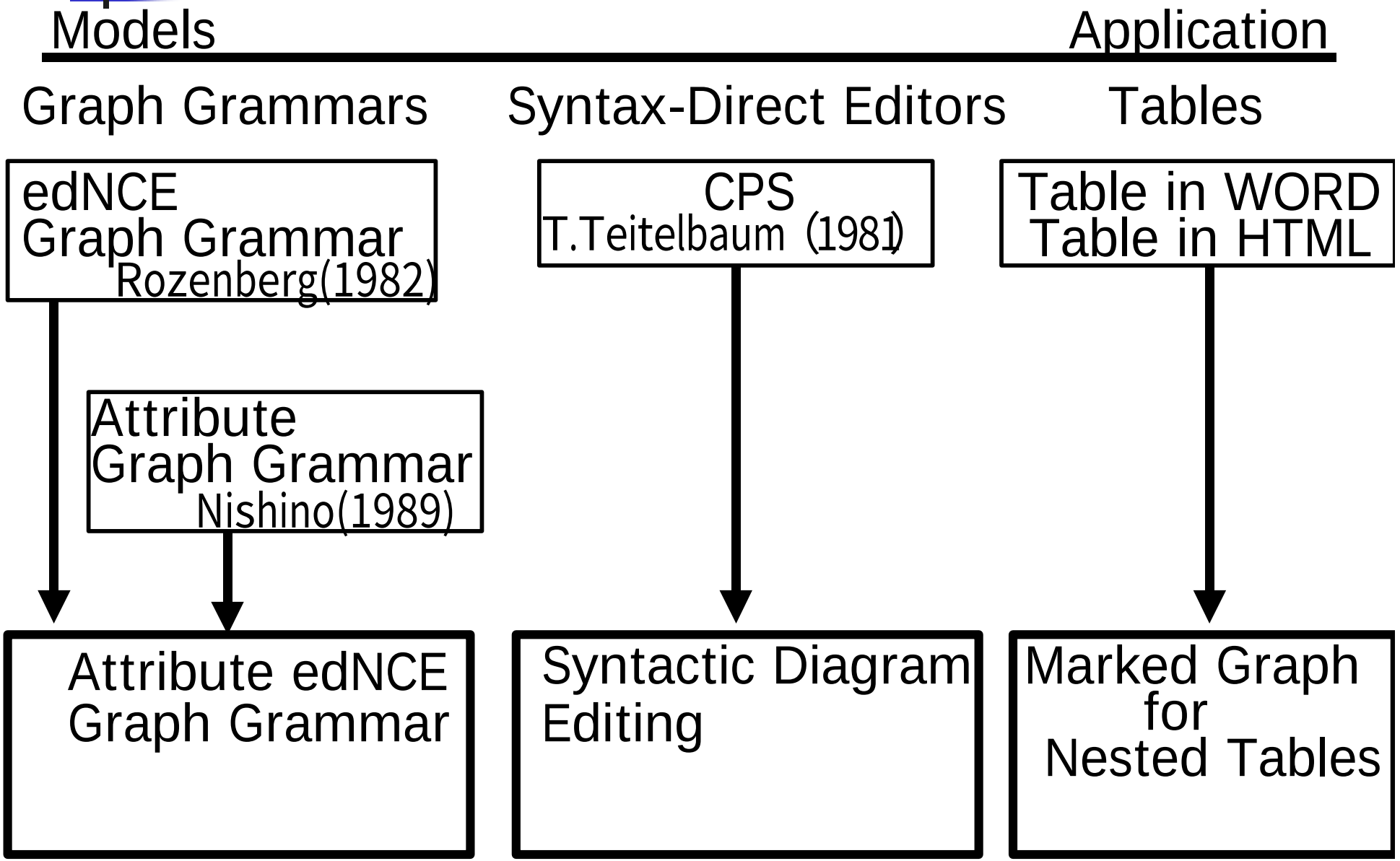
6. Conclusion

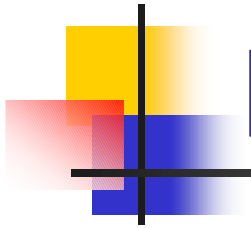
1. Introduction

位置づけ

	Flow Chart	Program Specification
Diagram	Hierarchical Diagram	Nested Diagram
		program name subtitle : library code : version : author : original release :
Graph	Attribute Tree	Attribute Marked Tree (ICSE98)
Graph Grammar	Attribute NCE CFGG (COMPSAC96)	Attribute NCE CFGG (IFIP2000)
Editor Command	O (COMPSAC96)	This paper

Back Ground

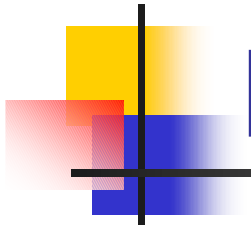

Models
Application
Graph Grammars
Syntax-Direct Editors
Tables
edNCE
Graph Grammar
Rozenberg(1982)
CPS
T.Teitelbaum (1981)
Table in WORD
Table in HTML
Attribute
Graph Grammar
Nishino(1989)
Attribute edNCE
Graph Grammar
Syntactic Diagram
Editing
Marked Graph
for
Nested Tables



Related Works

- Application of **Graph Grammar** were developed such as **DIAGEN**, **IPSEN** and **APPLIGRAPH**.

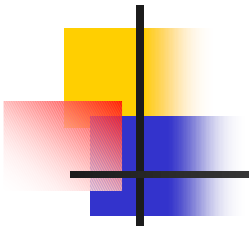
- Related works for **syntactic editing methods are CPS, DIAGEN** and so on.



Motivation

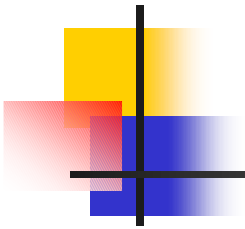
- A tabular form is formulated by the graph grammar.
- It’s necessary to make **syntactic editing method** by the graph grammar.

- To investigate effectiveness of formal methods of table editing.



Purpose

- Constructing the tabular form syntactic editing mechanisms **(Insert ,Add, Delete)** based on attribute graph grammar.
- We make composite production copies for the easy editing manipulation.



Results

- The definitions of the syntactic editing methods based on the attribute edNCE graph grammars for the tabular forms
  - Insert manipulation
    (274 composite productions)
  - Add manipulation
    (274 composite productions)
  - Delete manipulation
- The example of the insertion of the Item

2. Preliminaries

2. Preliminaries

Program Specification Hiform [10]

(a program specification language)

- 17 type of Forms based on ISO6592
- A collection of tabular forms

program name : Hanoi_main **subtitle :** hanoi	A General document
library code : cs - 2000 - 01	**version :** 1.0
author : Kiyonobu Tomiyama	**original release :** 1999/12/22
approve :	**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

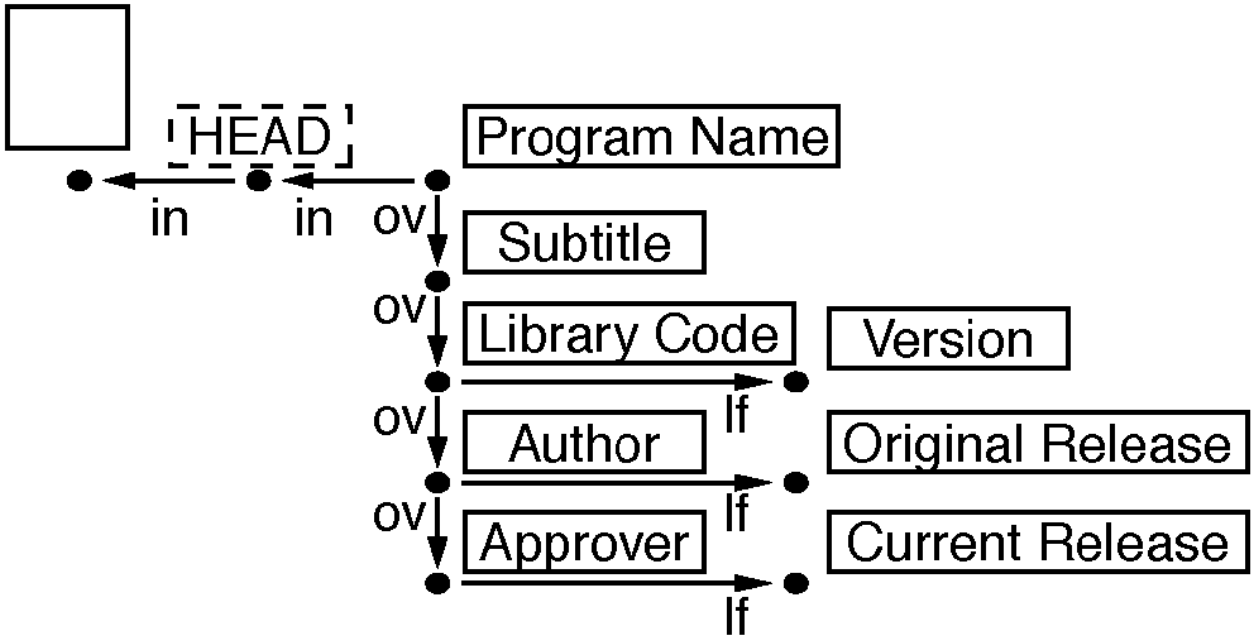
```
hanoi(5, A, B, C)
```

2. Example of Output

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

Tabular form and its corresponding graph

Program name : hanoi	
Subtitle :	
Library code : cs-2000-02	Version : 1.1
Author : K. Tomiyama	Original release :2000/6/10
Approver :	Current release :2000/10/1



2.1 An Attribute edNCE Graph Grammar

Definition

An attribute edNCE Graph Grammar：AGG=
＜G,Att,F＞

G=（Σ,Δ,Γ,Ω,P,S）：Underlying graph grammar of AGG

Att= $\dot{\cup}_{Y \in V}$ Att（Y）　　(Att(Y)=Inh(Y)∪Syn(Y))

F= $\dot{\cup}_{p \in P}$ F_p is the set of Semantic rules of AGG

edNCE Graph Grammar[6]

Definition

edNCE graph grammar：$G=(\Sigma,\Delta,\Gamma,\Omega,P,S)$

Σ：The alphabet of node labels

$\Delta \subseteq \Sigma$：The alphabet of terminal node labels

Γ：The alphabet of edge labels

$\Omega \subseteq \Delta$： The alphabet of final edge labels

P： The finite set of productions

$S \in \Sigma - \Delta$： The initial nonterminal

edNCE Graph Grammar (continued)

production $p : X \rightarrow (D, C)$

$X \in \Sigma - \Delta$

$(D, C) \in GRE_{\Sigma, \Gamma}$

$D \in GR_{\Sigma, \Gamma}$

$C \subseteq \Sigma \times \Gamma \times \Gamma \times V_D \times \{in, out\}$

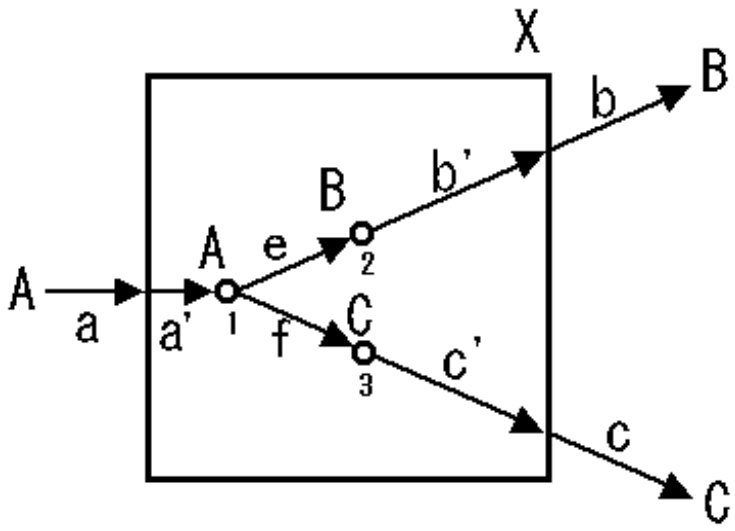
: connection relation

□

Production

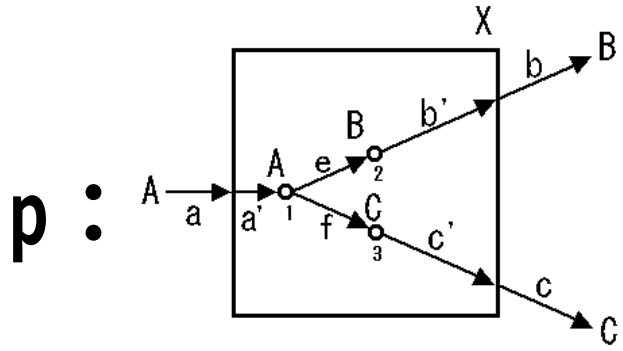
Example:

p :

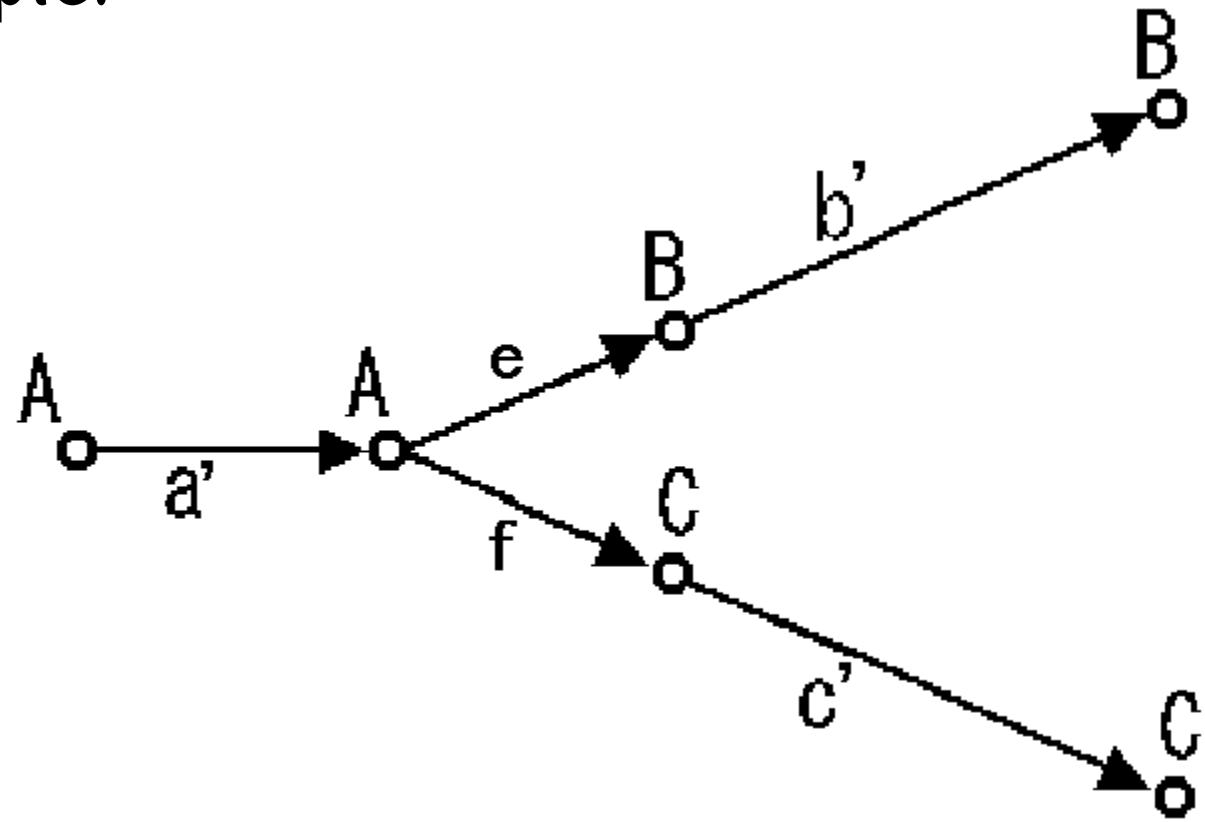


C={(A,a/a',1,in),(B,b/b',2,out),(C,c/c',3,out)}

Derivation



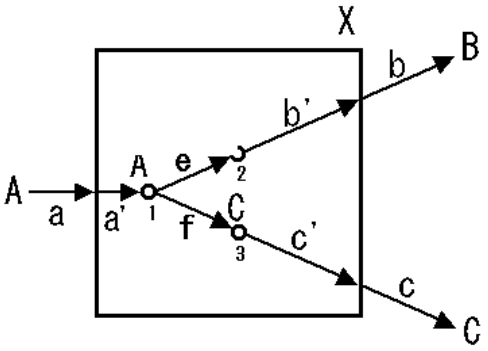
Example:

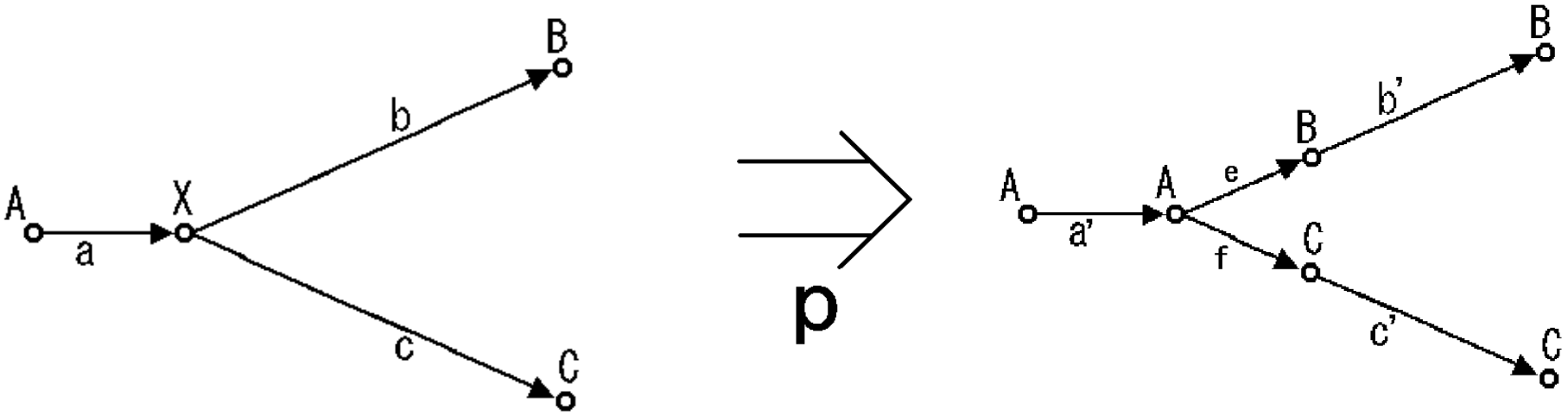


Derivation (continued)

Example:

p :





2.2 COMPOSITION OF PRODUCTION COPIES [4]

Definition

$G=(\Sigma,\Delta,\Gamma,\Omega,P,S)$: edNCE-CFGG

$p_1:X_1\rightarrow(D_1,C_1)$, $p_2:X_2\rightarrow(D_2,C_2)$: production copy of G

X_2 exists in node labels of D_1

The <u>composite production copy</u> $p:X_1\rightarrow(D,C)$ is defined as follows:

$D=D_1-\{X_2\}\cup D_2$

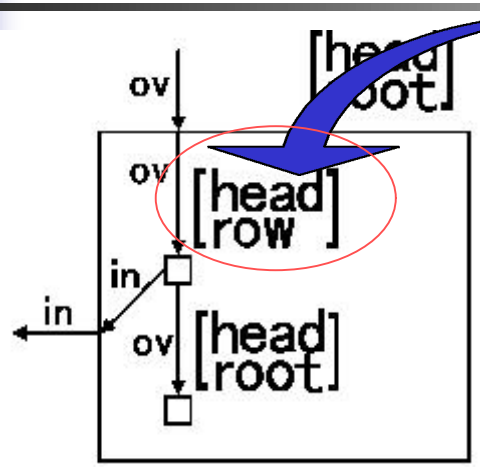
$C=\{(\sigma,\beta/\gamma,\omega,d)\in C1 \mid \omega\in V_{D1}-V_{X2}\}$

$\cup\{(\sigma,\beta/\delta,y,d) \mid \exists\,\omega\in V_{X2}\,\exists\,\gamma\in\Gamma,(\sigma,\beta/\gamma,\omega,d)\in C_1,$
$(\sigma,\gamma/\delta,y,d)\in C_2\}$

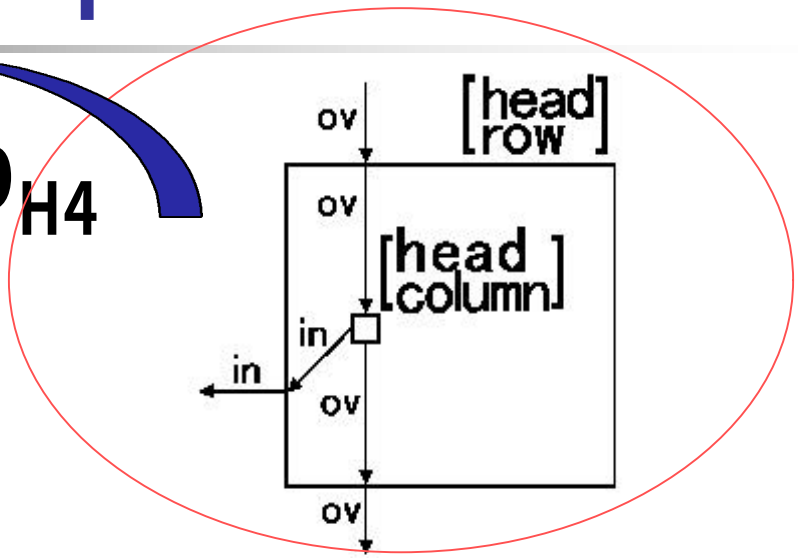
Denoted by $p1\circ p2$

Example : Composition of Production Copies

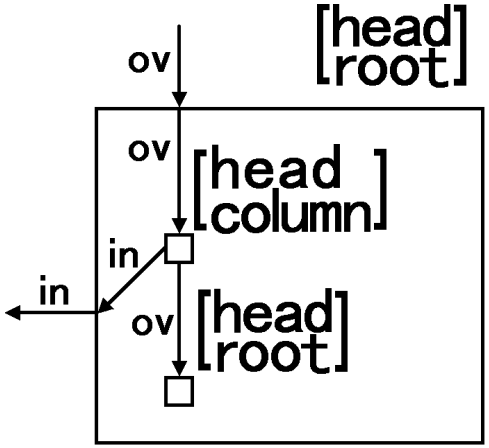
p_{H2}



p_{H4}



$p_{H2} \circ p_{H4}$



2.3 Confluence Property [6]

Definition

An edNCE graph grammar $G=(\Sigma,\Delta,\Gamma,\Omega,P,S)$ is <u>confluent</u> if the following holds for every sentential form H of G:

If $H \Rightarrow_{u1,p1} H1 \Rightarrow_{u2,p2} H12$ and $H \Rightarrow_{u2,p2} H1 \Rightarrow_{u1,p1} H21$ $(p1,p2 \in P)$ are derivation of G with $u1,u2 \in VH$ and $u1 \neq u2$,

then H12=H21

□

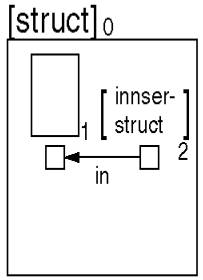
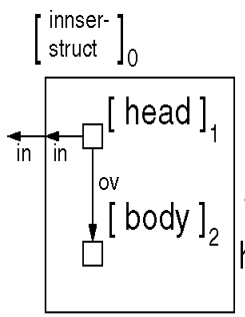
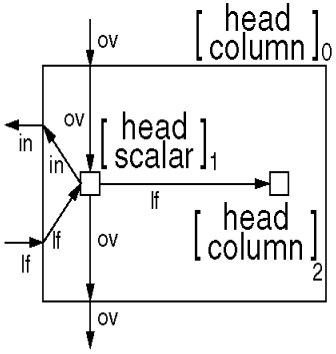
2.4 HNGG [10]

Hiform Nested Graph Grammar

HNGG=<G_N, A_N, F_N,>

Production : 280

Attribute rule : 1248

p_1	 x(1) = x(0) y(1) = y(0) x(2) = x(0) y(2) = y(0) width(0) = width(2) height(0) = height(2)
p_2	 x(1) = x(0)+Mleft y(1) = y(0)+Mtop x(2) = x(0)+Mleft y(2) = y(0)+height(1)+Mcen width(0) = max(width(1),width(2)) height(0) = height(1)+height(2) +Mtop+Mcen+Mbottom
p_{H5}	 x(1) = x(0) x(2) = x(0)+width(1)+HSh y(1) = y(0) y(2) = y(0) width(0) = width(1)+width(2)+HSh height(0) = max(height(1), height(2))

3 Editing of Nested Tabular Form

3. 1 Production Instance

Production Instance : (ω, p_i, H'_{pi})

$\omega \in VD_{i-1}$: a node removed during the derivation

$D_{i-1} \Longrightarrow_{pi} D_i$

$p_i \in P$: a production

H'_{pi} : an embedded graph isomorphic H_{pi} during

$D_{i-1} \Longrightarrow_{pi} D_i$

$D_{i-1} \overset{\omega H'pi}{\underset{pi}{\Longrightarrow}} D_i$: D_{i-1} is directly derived D_i by applying the (ω, p_i, H'_{pi})

3.2 Syntactic Insertion

3.2 Syntactic Insertion

Definition (insertable)

For the derivation sequence

$$D_0 \overset{\omega_{i-1}\,H'_{pi-1}}{\underset{p_{i-1}}{\Rightarrow}} \cdots \overset{\omega_{i-1}\,H'_{pi-1}}{\underset{p_{i-1}}{\Rightarrow}} D_{i-1} \overset{\omega\,H'_{pi}}{\underset{p_i}{\Rightarrow}} D_i \overset{\omega_{i+1}\,H'_{pi+1}}{\underset{p_{i+1}}{\Rightarrow}} \cdots \overset{\omega_n\,H'_{pn}}{\underset{p_n}{\Rightarrow}} D_n \quad (p_j: Xp_j \rightarrow (Hp_j, Cp_j), 1 \leqq j \leqq n),$$

Production q is <u>insertable</u> (for pi) $\overset{def}{\Leftrightarrow}$

Instance (ω, q, H'_q) $(q = Xq \rightarrow (Hq, Cq) \in P)$ such that $D_{i-1} \overset{\omega\,H'_q}{\underset{q}{\Rightarrow}} Q$,

$D_{i-1} \overset{\omega\,H'_q}{\underset{q}{\Rightarrow}} Q \overset{\omega'\,H'_{pi}}{\underset{p_i}{\Rightarrow}} D'_i \overset{\omega_{i+1}\,H'_{pi+1}}{\underset{p_{i+1}}{\Rightarrow}} \cdots \overset{\omega_n\,H'_{pn}}{\underset{p_n}{\Rightarrow}} D'_n$ exists. □

If a production $q \in P_N$ is insertable for p_i , then an instance sequence S is obtained by <u>insertion</u> of an instance (ω, q, H'_q) into an instance sequence

$((\omega_1, p_1, H'_{p1}), \cdots, (\omega_i, p_i, H'_{pi}), \cdots, (\omega_n, p_n, H'_{pn}))$

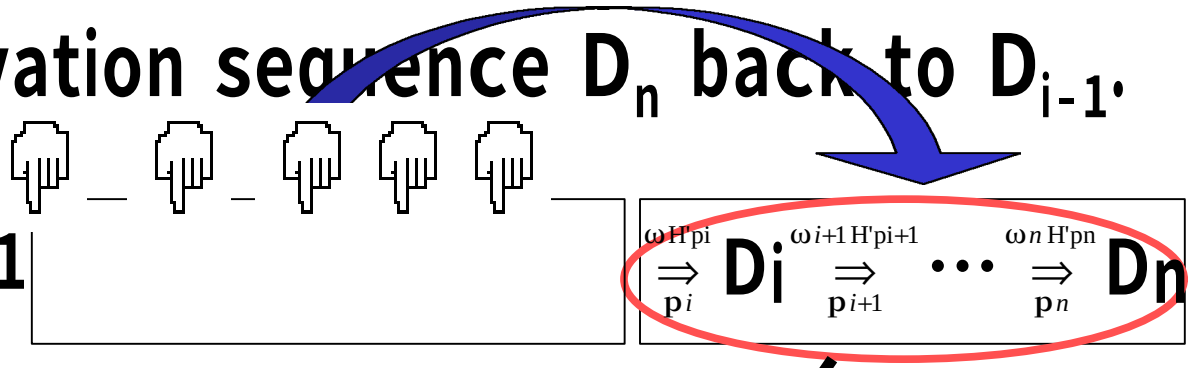
makes an instance sequence

$\overset{def}{\Leftrightarrow}$

$S=((\omega_1, p_1, H'_{p1}), \cdots, (\omega_{i-1}, p_{i-1}, H'_{pi-1}), (\omega, q, H'_q),$
$(\omega'_i, p_i, H'_{pi}), \cdots, (\omega'_n, p_n, H'_{pn}))$

There is the instance sequence S as follows.

1. Trace the derivation sequence D_n back to D_{i-1}.



2. Apply the instance (ω, q, H'_q) to D_{i-1}, and get the resultant graph Q .

$$D0 \overset{\omega 1\, H'p1}{\underset{p1}{\Rightarrow}} \cdots \overset{\omega i-1\, H'pi-1}{\underset{p\,i-1}{\Rightarrow}} Di\text{-}1 \overset{\omega H'q}{\underset{q}{\Rightarrow}} Q$$

3. Apply the instance sequence $(\omega'_i, p_i, H'_{pi}), \cdots, (\omega'_n, p_n, H'_{pn})$ to Q, and get the resultant graph D'_n.

$$D0 \overset{\omega 1\, H'p1}{\underset{p1}{\Rightarrow}} \cdots \overset{\omega i-1\, H'pi-1}{\underset{p\,i-1}{\Rightarrow}} Di\text{-}1 \overset{\omega H'q}{\underset{q}{\Rightarrow}} Q \overset{\omega' H'pi}{\underset{p\,i}{\Rightarrow}} D'i \overset{\omega i+1\, H'pi+1}{\underset{p\,i+1}{\Rightarrow}} \cdots \overset{\omega n\, H'pn}{\underset{p\,n}{\Rightarrow}} D'n$$

Definition

A graph H' is obtained by syntactic insertion of a e graph A at edge x in a target graph H.

$\overset{def}{\Leftrightarrow}$

1. A composite production copy q for the graph A and an edge x exists.
2. There exists an instance sequence i_H for H.

An instance sequence S is obtained by insertion of q into i_H.

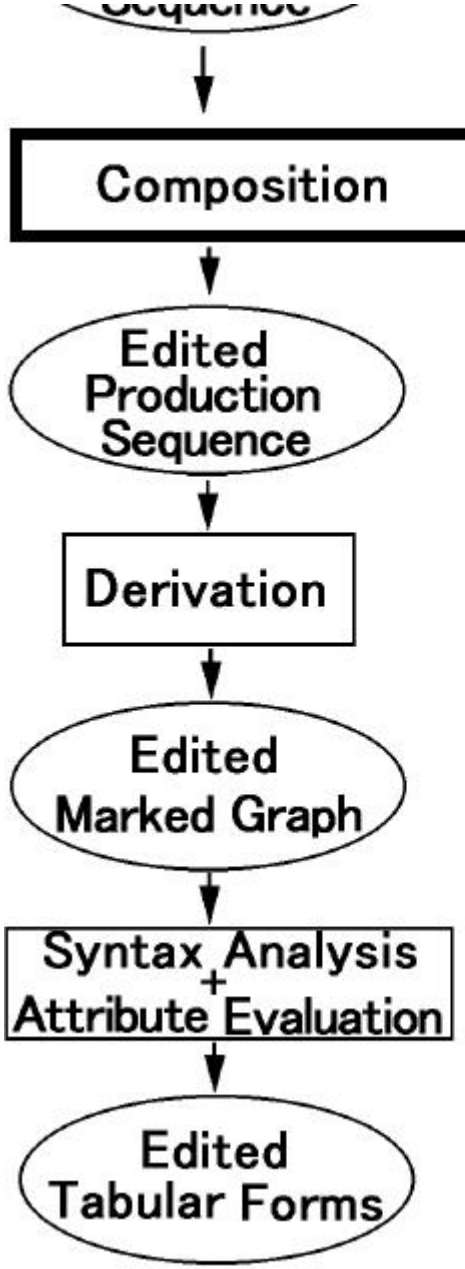
3.The graph H' is derived by the S.

Theorem 3.2

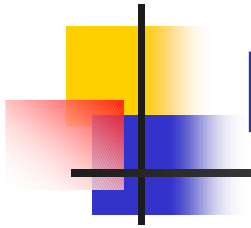
Let H be the graph obtained from G by the insertion of nodes a and b at an edge x and edge y respectively in this order, in HNGG.Let H' be the graph obtained from G by the insertion of nodes b and a at an edge x and edge y respectively in this order, in HNGG.Then, H=H'. □

Proof

HNGG has confluence property. □



A process flow for an insertion of Hiform editing system

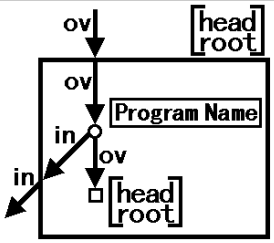
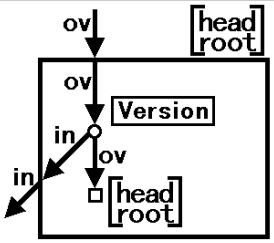
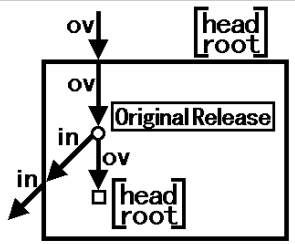
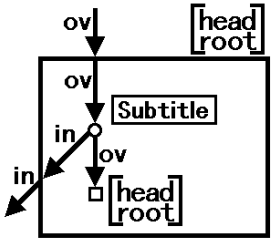
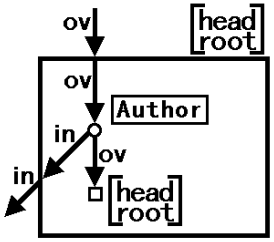
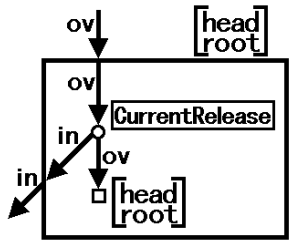
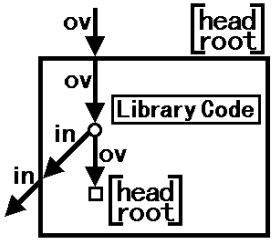
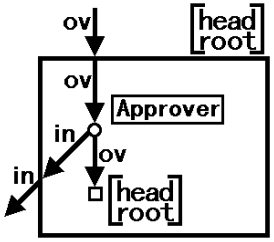


Definition

In the same manner as the editing by the instance for a production,we can further define insertable *by composite production copy.*

□

Composite Productions for Insert

$((P_{H2} \circ P_{H4}) \circ P_{H6}) \circ P_{H7}$		$((P_{H2} \circ P_{H4}) \circ P_{H6}) \circ P_{H10}$		$((P_{H2} \circ P_{H4}) \circ P_{H6}) \circ P_{H13}$	
$((P_{H2} \circ P_{H4}) \circ P_{H6}) \circ P_{H8}$		$((P_{H2} \circ P_{H4}) \circ P_{H6}) \circ P_{H11}$		$((P_{H2} \circ P_{H4}) \circ P_{H6}) \circ P_{H14}$	
$((P_{H2} \circ P_{H4}) \circ P_{H6}) \circ P_{H9}$		$((P_{H2} \circ P_{H4}) \circ P_{H6}) \circ P_{H12}$			

3.3 Syntactic Addition

Syntactic Addition

Definition (addable)

For the derivation sequence

$$D_0 \underset{p_1}{\Rightarrow} \cdots \underset{p_{i-1}}{\Rightarrow} \boxed{D_{i-1}} \underset{p_i}{\Rightarrow} D_i \boxed{\underset{p_{i+1}}{\Rightarrow} D_{i+1} \underset{p_{i+2}}{\Rightarrow} \cdots \underset{p_n}{\Rightarrow} D_n}$$

$(p_j: X_{p_j} \rightarrow (H_{p_j}, C_{p_j}), 1 \leq j \leq n)$,

Production $q: X_q \rightarrow (H_q, C_{pq})$ is <u>addable</u> (for p_i) :

1. $\boxed{D_{i-1} \underset{q}{\overset{\omega H'q}{\Rightarrow}} Q}$ and $D_{i-1} \underset{q}{\overset{\omega H'q}{\Rightarrow}} Q \boxed{\underset{p_{i+1}}{\overset{\omega_{i+1} H'p_{i+1}}{\Rightarrow}} D'_{i+1} \underset{p_{i+2}}{\overset{\omega_{i+2} H'p_{i+2}}{\Rightarrow}} \cdots \underset{p_n}{\overset{\omega_n H'p_n}{\Rightarrow}} D'_n}$

2(a). $X_q = X_{p_i}$

(b). $V_{Hq} = V_{H'p_i} + \{u\}$

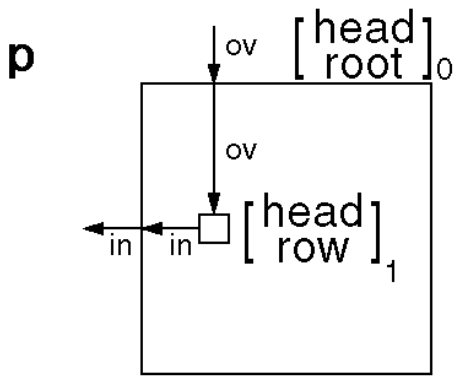
(c). If f and g are isomorphic mappings such that

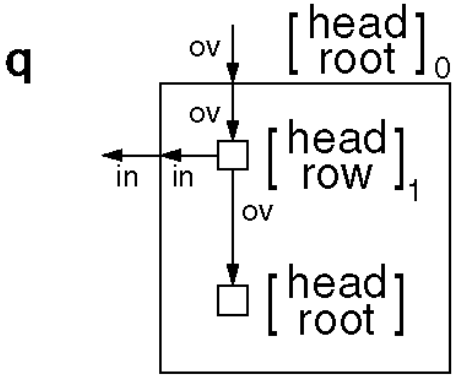
$f: V_{H'p_i} \rightarrow V_{Hp_i}$

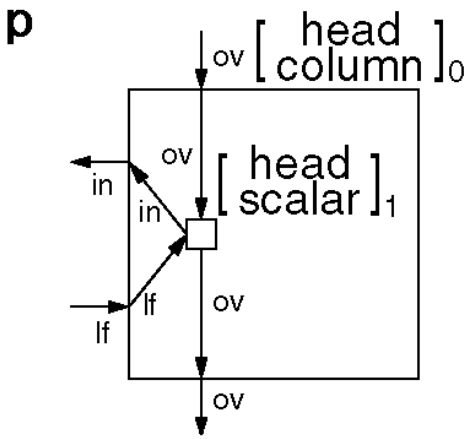
$g: V_{H'p_i} + \{u\} \rightarrow V_{H'q}$

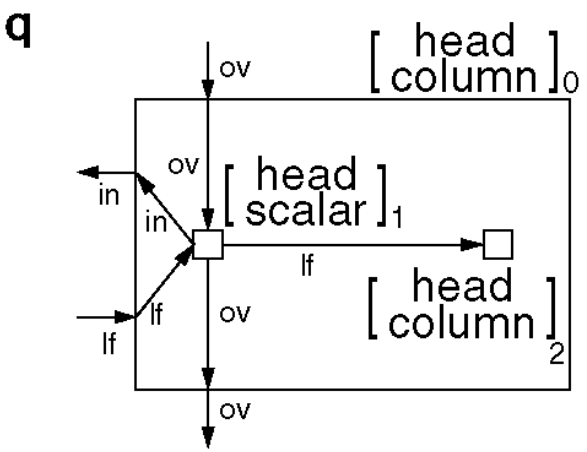
then $(\sigma, \beta / \gamma, y, d) = (\sigma, \beta / \gamma, g(y), d)$ □

Example


p
ov
[head root]0
ov
in
in
[head row]1

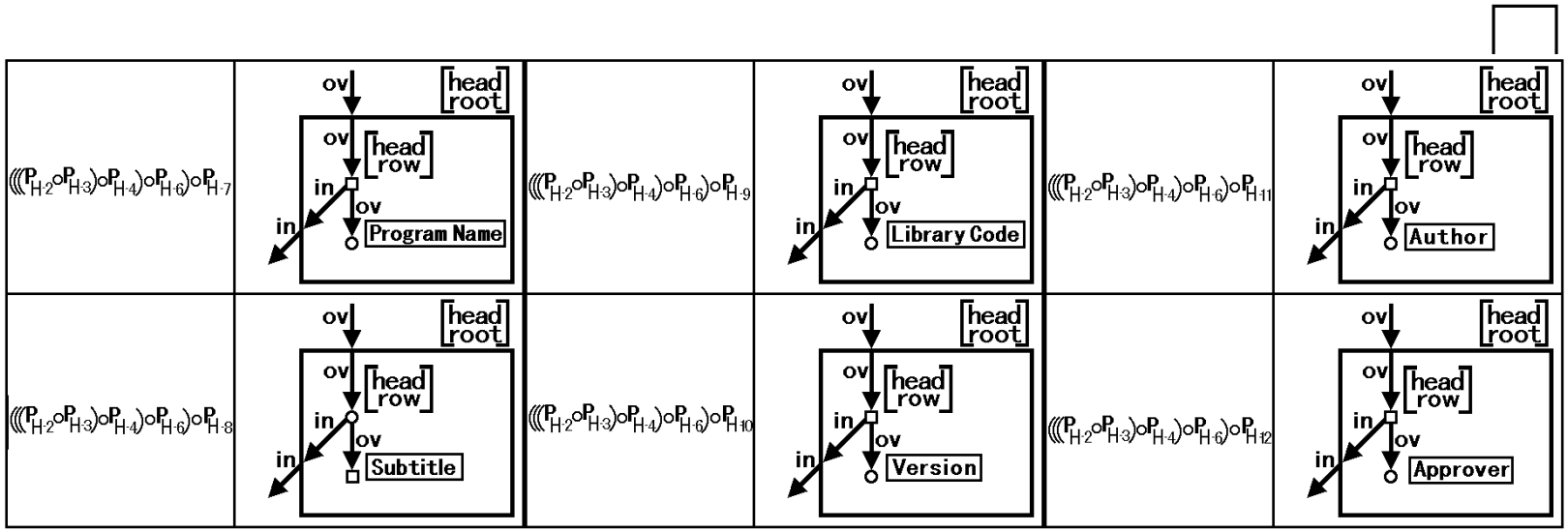

q
ov
[head root]0
ov
in
in
[head row]1
ov
[head root]


p
ov
[head column]0
ov
in
in
[head scalar]1
lf
lf
ov
ov

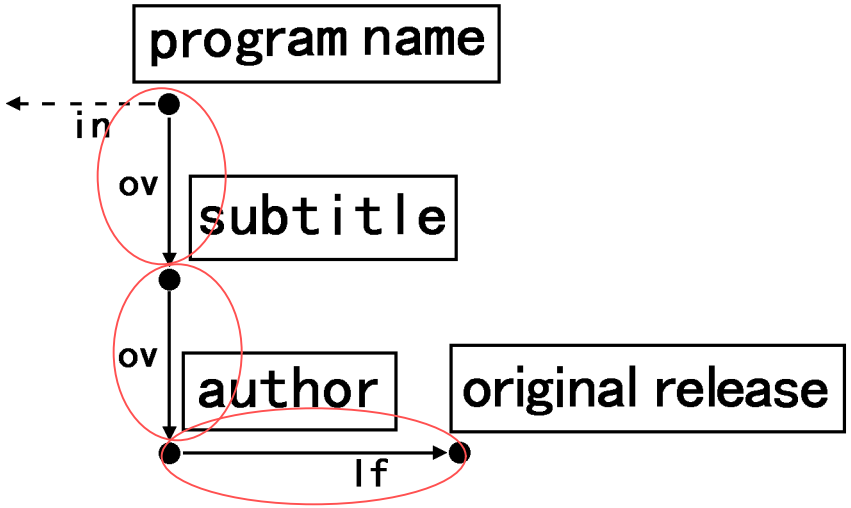

q
ov
[head column]0
ov
in
in
[head scalar]1
lf
lf
lf
ov
[head column]2
ov

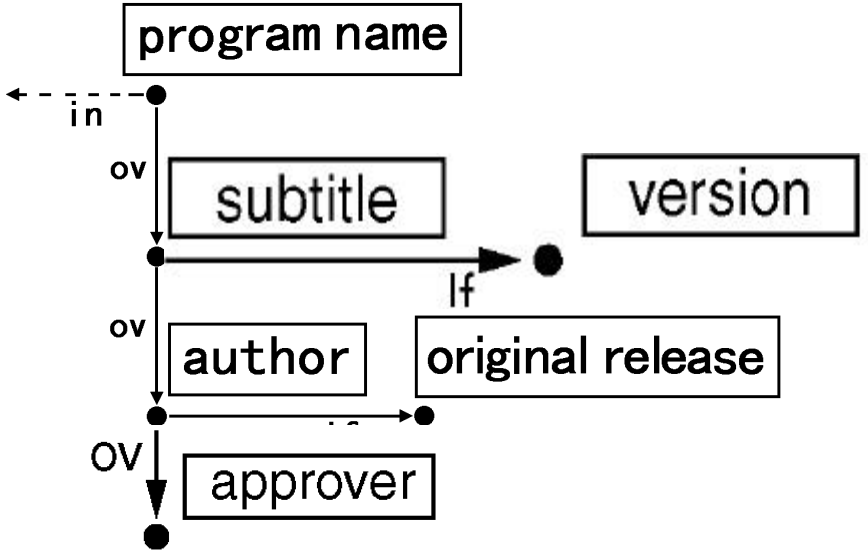
Composite Productions for add Definition

In the same manner as the editing by the instance for a production,we can further define addable *by composite production copy.*



Insert and Add





3.4 Syntactic Deletion of Item

3.4 Syntactic Deletion of Item

Definition (deletable)

For the derivation sequence $D_0 \overset{\omega 1\, H'p1}{\underset{p1}{\Rightarrow}} \cdots \overset{\omega k\, H'k}{\underset{pk}{\Rightarrow}} F \overset{\omega\, H'p}{\underset{p}{\Rightarrow}} D_p \overset{\omega l\, H'l}{\underset{pl}{\Rightarrow}} \cdots \overset{\omega n\, H'pn}{\underset{pn}{\Rightarrow}} D_n$,

The graph that D_p has node $u \in VD$ for the first time.

Node u is not rewritten by any production after that.

Production $p=Xp \rightarrow (Dp,Cp) \in PN$ is <u>deletable</u> if one of the following Assumptions 1-3 is met

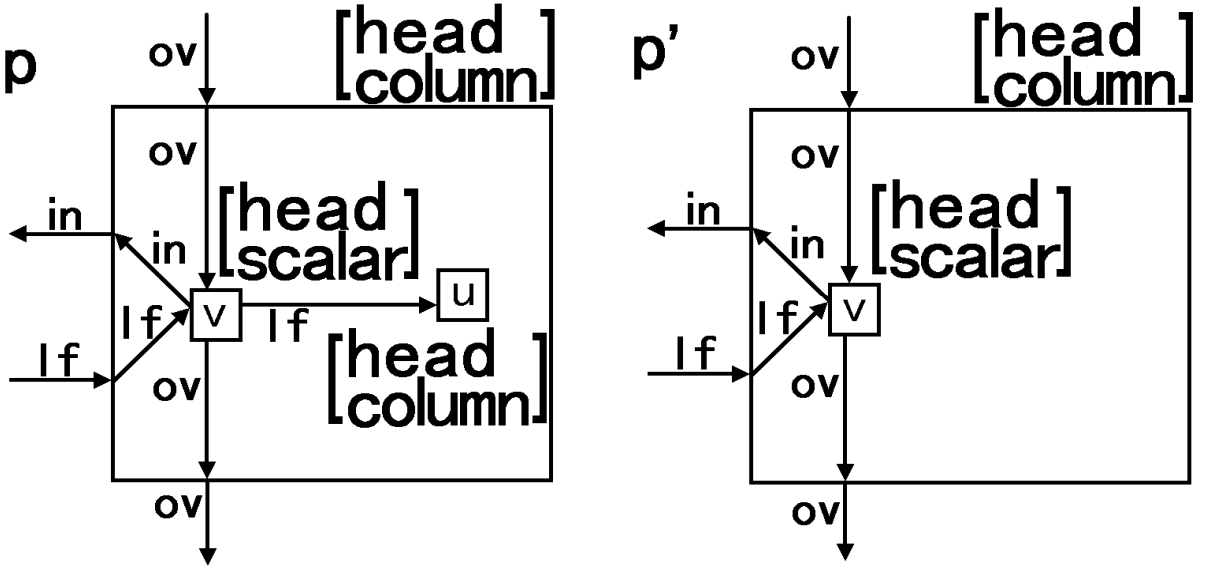
Assumption 1

For $p \in P_N$, $\exists\ p'$: $X_p \rightarrow (D_p, C_p) \in P_N$ s.t.

1. $X_{p'} = X_p$
2. $H_{p'} \equiv H'_p - \{u\}$
3. f、g: isomorphic mappings, $f: V_{H'p} \rightarrow V_{Hp}$、$g: V_{Hp-\{f(u)\}} \rightarrow V_{Hp'}$

then $(\sigma, \beta / \gamma, y, d) = (\sigma, \beta / \gamma, g(y), d)$

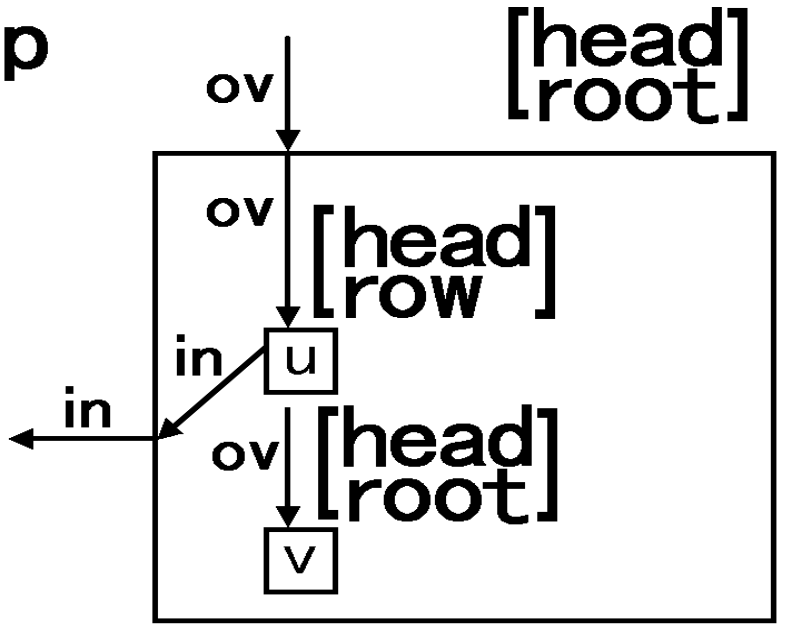
For example:



Assumption 2

$$V_{H'p} = \{u,v\} \ , \ X_{H'p} = H'p(v)$$

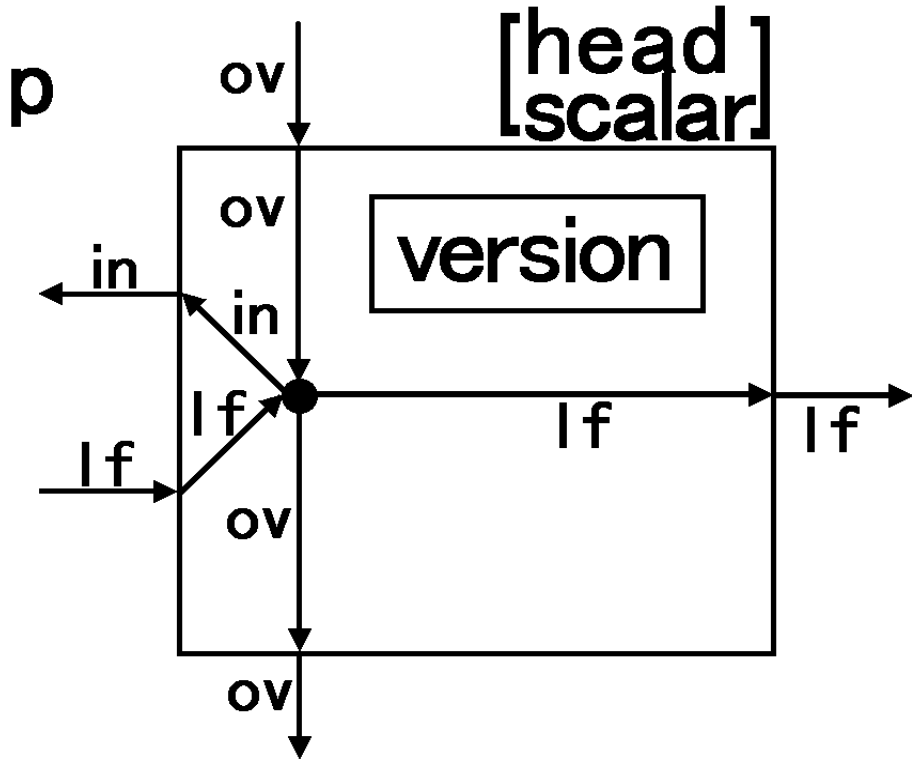
For example:



Assumption 3

$\omega_j \notin H'p$, $1 \leqq j \leqq n$

For example:



□

The case of Assumption1

If a production $p \in P_N$ is deletable, then an instance sequence S is obtained by deletion of an instance (ω, p, H'_p) from a instance sequence

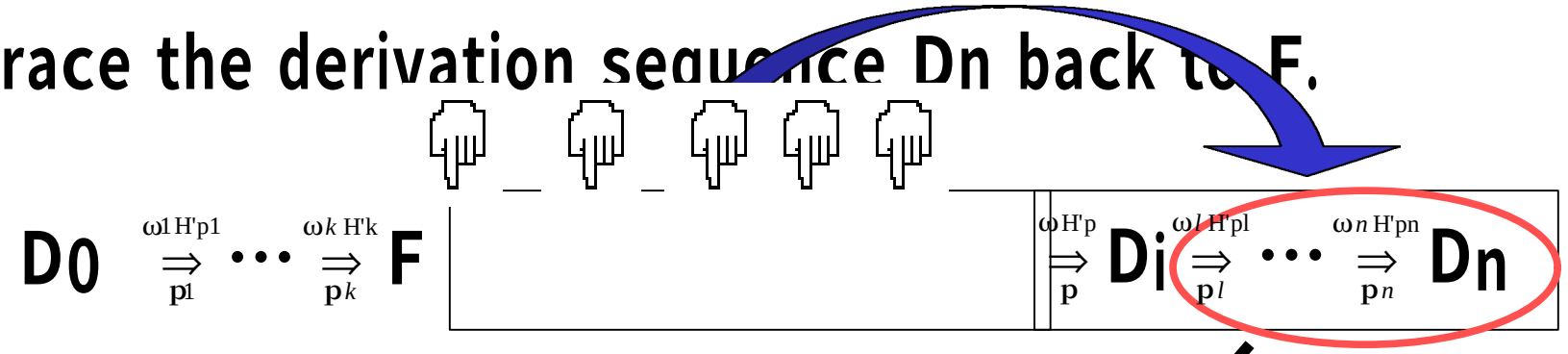
$((\omega_1, p_1, H'_{p1}), \cdots, (\omega_k, p_k, H'_{pk}), (\omega, p, H'_p), (\omega_l, p_l, H'_l), \cdots, (\omega_n, p_n, H'_{pn}))$

$\overset{def}{\Leftrightarrow}$

$S=((\omega_1, p_1, H'_{p1}), \cdots, (\omega_k, p_k, H'_{pk}), (\omega, p', H'_{p'}), (\omega'_l, p_l, H'_l), \cdots, (\omega'_n, p_n, H'_{pn}))$

There is the S as follows.

1. Trace the derivation sequence Dn back to F.



2. Apply the instance $(\omega, p', H'_{p'})$ to F, and get the resultant graph D'p.

$$D0 \overset{\omega 1\, H'p1}{\underset{p1}{\Rightarrow}} \cdots \overset{\omega k\, H'k}{\underset{pk}{\Rightarrow}} F \overset{\omega\, H'p'}{\underset{p'}{\Rightarrow}} D\,'p$$

3. Apply the instance sequence $(\omega'_1, p_1, H'_1), \cdots, (\omega'_n, p_n, H'_{pn})$ to D'p, and get the resultant graph D'n.

$$D0 \overset{\omega 1\, H'p1}{\underset{p1}{\Rightarrow}} \cdots \overset{\omega k\, H'k}{\underset{pk}{\Rightarrow}} F \overset{\omega\, H'p'}{\underset{p'}{\Rightarrow}} D\,'p \overset{\omega l\, H'l}{\underset{pl}{\Rightarrow}} \cdots \overset{\omega n\, H'pn}{\underset{pn}{\Rightarrow}} D\,'n$$

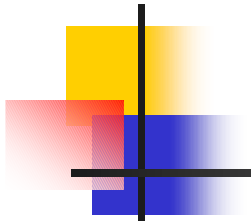
Definition

A graph *H'* is <u>obtained by syntactic deletion</u> of a node *A* from a graph *H*

①A production *q* having a node *A* on the right hand side exists.

②Let i_H is an instance sequence for *H*. Let i_q is a production instance for *q* in i_H. The *q* is deletable in i_H. An instance sequence *S* is obtained by deletion of i_q from i_H.

③The graph *H'* is derived by *S*.



Theorem 3.4

Let H be the graph obtained from G by the deletion of nodes a and b in this order, in HNGG.Let H' be the graph obtained from G by the deletion of nodes b and a in this order, in HNGG.Then, H=H'. □

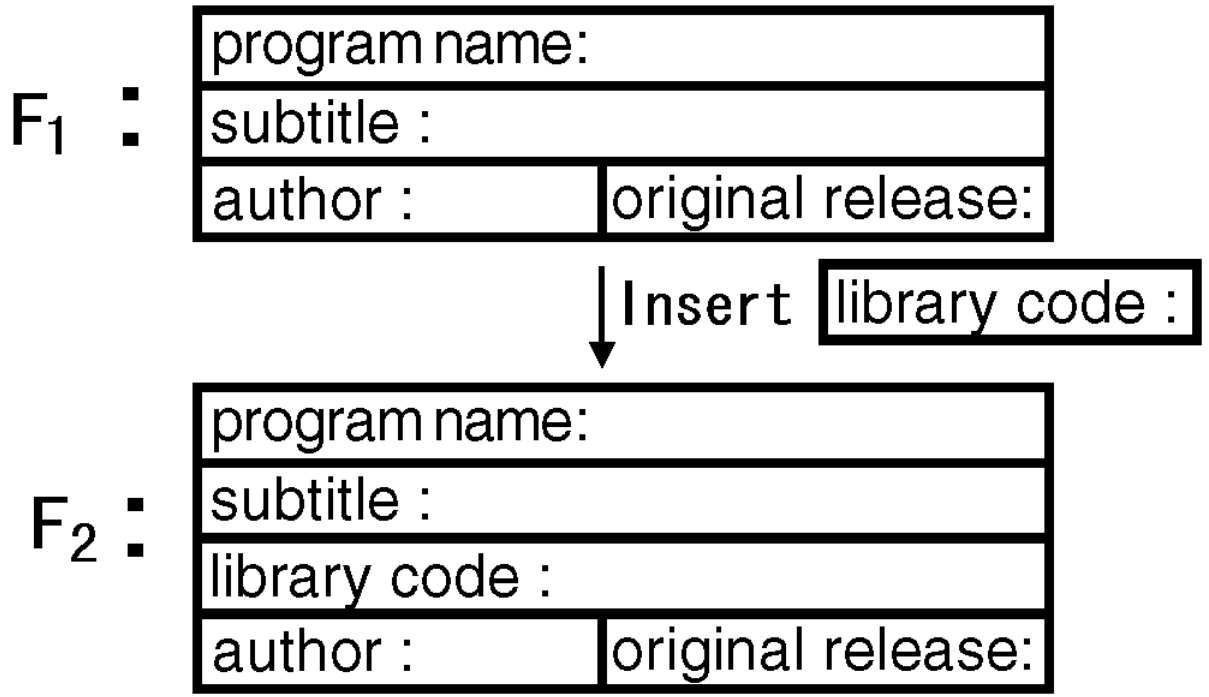
Proof

HNGG has confluence property. □

4 Example: Insertion Process

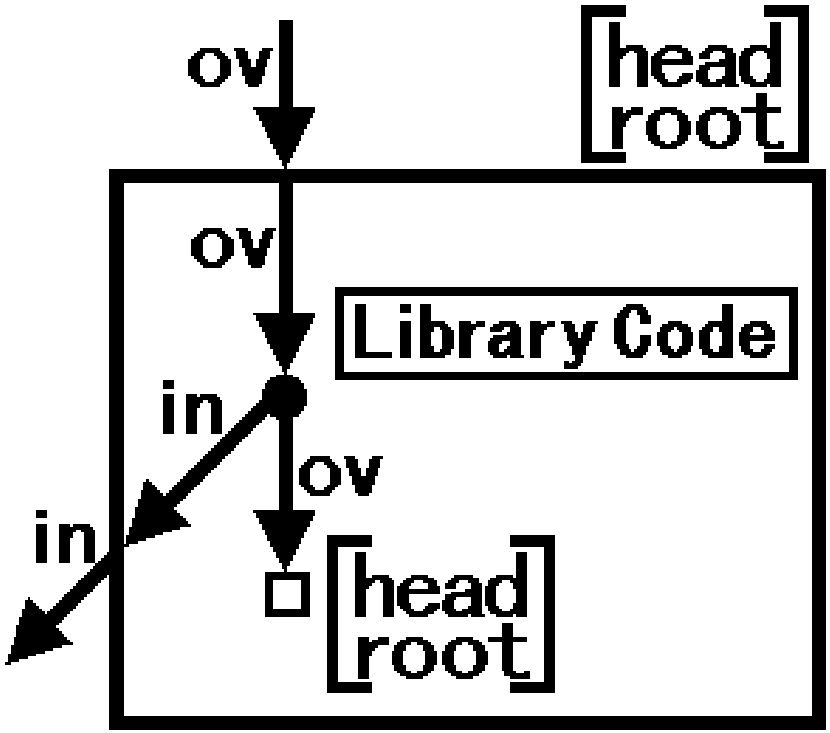
4 Example: Insertion Process

Insertion of library code between the 2nd and the 3ed branches of F_1



Step1. It makes the composite production copy for the use of the insertion.

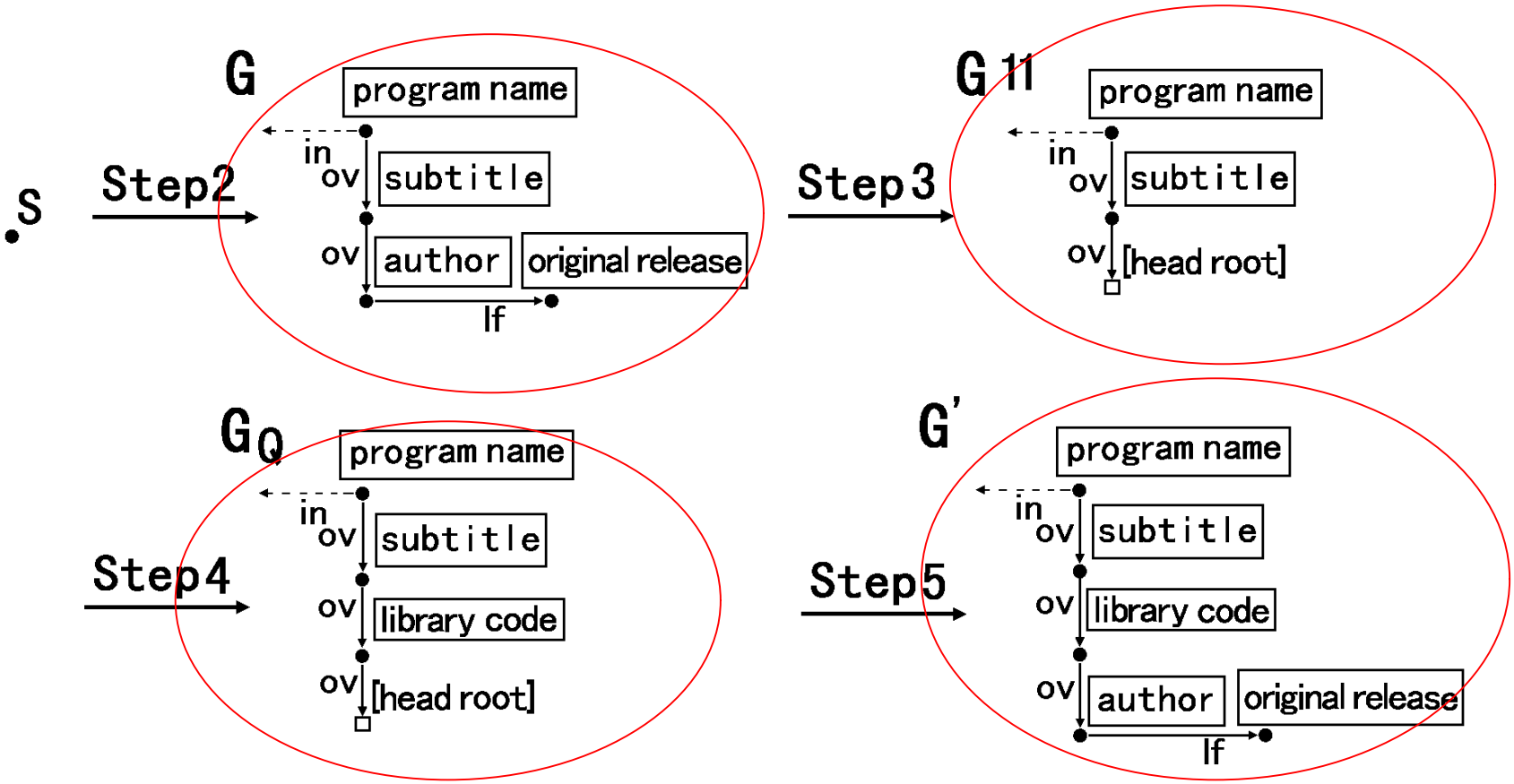
$P_Q = (((P_{H2} \bigcirc P_{H4}) \bigcirc P_{H5}) \bigcirc P_{H6}) \bigcirc P_{H9}) \bigcirc P_{H10}$



$$D_1 : G_0 \overset{\omega 1\ H'p1}{\underset{p1}{\Rightarrow}} G_1 \overset{\omega 2\ H'p2}{\underset{p2}{\Rightarrow}} G_2 \overset{\omega H1\ H'pH1}{\underset{pH1}{\Rightarrow}} G_3 \overset{\omega H2\ H'pH2}{\underset{pH2}{\Rightarrow}} G_4 \overset{\omega H4\ H'pH4}{\underset{pH4}{\Rightarrow}} G_5 \overset{\omega H6\ H'pH6}{\underset{pH6}{\Rightarrow}} G_6 \overset{\omega H7\ H'pH7}{\underset{pH7}{\Rightarrow}} G_7$$

$$\overset{\omega H2\ H'pH2}{\underset{pH2}{\Rightarrow}} G_8 \overset{\omega H4\ H'pH4}{\underset{pH4}{\Rightarrow}} G_9 \overset{\omega H6\ H'pH6}{\underset{pH6}{\Rightarrow}} G_{10} \overset{\omega H8\ H'pH8}{\underset{pH8}{\Rightarrow}} G_{11} \overset{\omega QH'pQ}{\underset{pQ}{\Rightarrow}} G_Q \overset{\omega H3\ H'pH3}{\underset{pH3}{\Rightarrow}} G_{12} \overset{\omega H4\ H'pH4}{\underset{pH4}{\Rightarrow}} G_{13}$$

$$\overset{\omega H5\ H'pH5}{\underset{pH5}{\Rightarrow}} G_{14} \overset{\omega H11\ H'pH11}{\underset{pH11}{\Rightarrow}} G_{15} \overset{\omega H6\ H'pH6}{\underset{pH6}{\Rightarrow}} G_{16} \overset{\omega H13\ H'pH13}{\underset{pH13}{\Rightarrow}} G$$

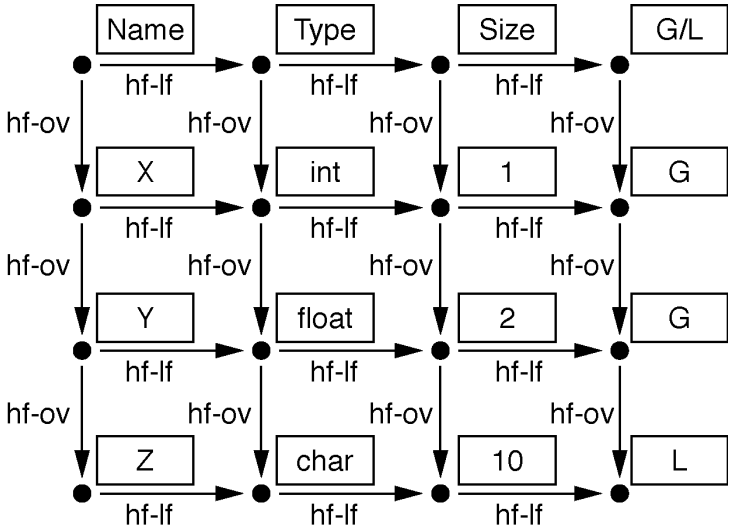


Insertion process

5. Editing of Tessellation Tabular Forms

Tessellation Tabular form and its corresponding graph

Name	Type	Size	G/L
X	int	1	G
Y	float	2	G
Z	char	10	L



5.1 HTGG [10]

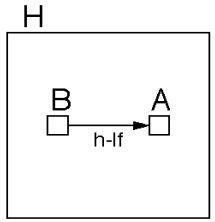
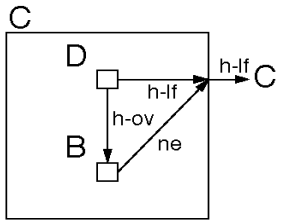
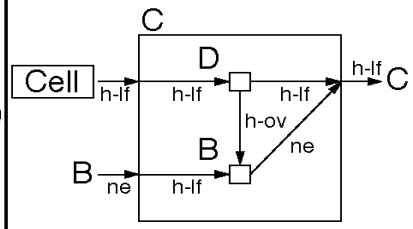
Hiform Tessellation Graph Grammar

HTGG=<GT, AT, FT,>

Underlying graph grammar

$G_T = (\Sigma_T, \Delta_T, \Gamma_T, \Omega_T, P_T, S_T)$

(context-sensitive edNCE graph grammar)

	Productions and Semantic Rules for Tessellation Forms (Horizontal Derivation 1)
p_2	 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))
p_8	 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)
p_{10}	 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)

5.2 Editing method of Tessellation Tabular Form

Definition (insertable)

Derivation sequence $D_0 \overset{\omega_1 H'p_1}{\underset{p_1}{\Rightarrow}} \cdots \overset{\omega_{i-1} H'p_{i-1}}{\underset{p_{i-1}}{\Rightarrow}} D_{i-1} \overset{\omega H'p_i}{\underset{p_i}{\Rightarrow}} D_i \overset{\omega_{i+1} H'p_{i+1}}{\underset{p_{i+1}}{\Rightarrow}} \cdots \overset{\omega_n H'p_n}{\underset{p_n}{\Rightarrow}} D_n$

$(p_j = X_{p_j} \rightarrow (H_{p_j} C_{p_j}), 1 \leqq j \leqq n)$ such that

$x \in D_n$ is a head node of a selected line

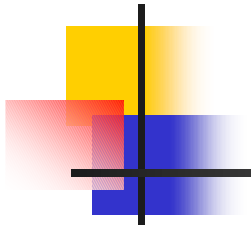
D_i is a graph in which x appears for the first time.

q is insertable for p_i if

(1) there is a production sequence $q = (q_1 \cdots q_m)$ such that

$D_{i-1} \overset{\omega H'q}{\underset{q}{\Rightarrow}} Q \overset{\omega' H'p_i}{\underset{p_i}{\Rightarrow}} D'_i \overset{\omega_{i+1} H'p_{i+1}}{\underset{p_{i+1}}{\Rightarrow}} \cdots \overset{\omega_n H'p_n}{\underset{p_n}{\Rightarrow}} D'_n$

(2) it is possible to apply any derivation sequences for D_i, then it is also possible to apply them for D'_i.



insertion command.

Definition

It prepares

q=(p17, p18, (p19, p20) ^{k}, p21, p22, p14, p15^{k}, p16) as the insertion command.

6.Conclusion

- We proposed syntactic editing methods for tabular forms, based on the attribute edNCE graph grammar.
  - **Insert manipulation (274 composite productions)**
  - **Add manipulation (274 composite productions)**
  - **Delete manipulation**
- Examples to apply editor methods were shown.
- We considered the editing methods of the tessellation tabular form.