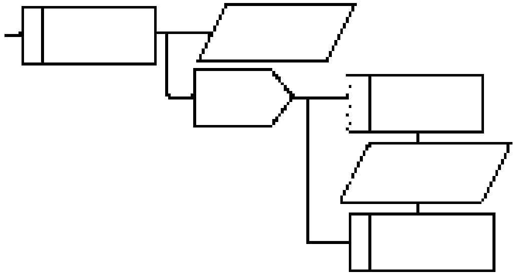
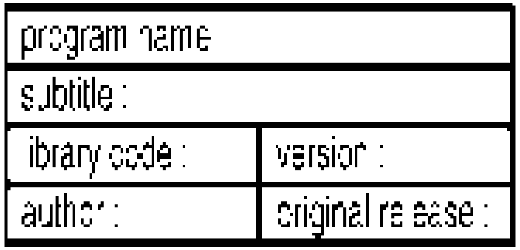
Syntactic Editing of Tabular Forms by Attribute edNCE Graph Grammars

Kiyonobu TOMIYAMA (Nihon Univ.)
Tomokazu ARITA (Nihon Univ.)
Kensei TSUCHIDA (Toyo Univ.)
Takeo YAKU (Nihon Univ.)

1. Introduction

Target

	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	○ (COMPSAC96)	This paper

Motivation

- Graph grammar was used to the formalism of the tabular forms.
- Uses an attribute for the layout.
  - It’s necessary to make syntactic editing method by an attribute graph grammar.

Back Ground

The Cornel Program Synthesizer (CPS)

(text-based editor) T.Teitelbaum (1981)

The graph editing by the graph grammar

Göttler (1986)

Attribute Graph Grammar Nishino(1989)

edNCE Graph Grammar Rozenberg(1996)

Our History

1996 Hichart Diagram Syntactic Editing
Command [COMPSAC9 6] (Anzai et al.)

1997 Program Specification Hiform
[APEC97] (Sugita et al,)

2000 Syntactic Processing of Diagrams by
Graph Grammars [IFIP WCC 2000]

This paper Tabular Diagram Editing Methods

Related Works

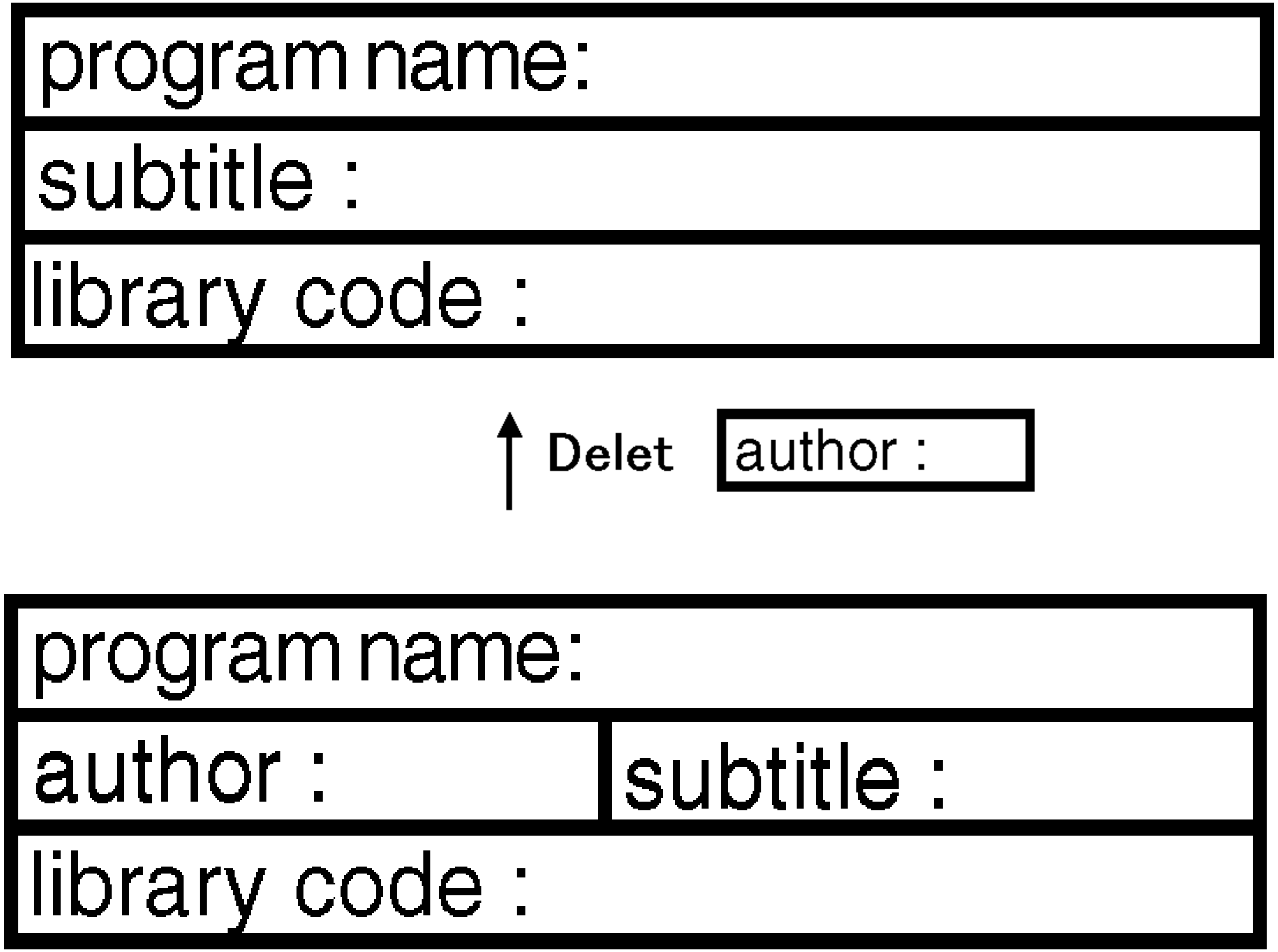
- Application of Graph Grammar were developed such as DIAGEN, IPSEN and APPLIGRAPH.
- Our project for graph processing was named KEYAKI-CASE2000.

Purpose

- Constructing the tabular form editing mechanisms based on graph grammar

Results

- The definitions of the editing methods based on the attribute edNCE graph grammars for the tabular forms
- The example of the insertion and the deletion of the Item



program name:

subtitle :

library code :

Delet

author :

program name:

author :

subtitle :

library code :

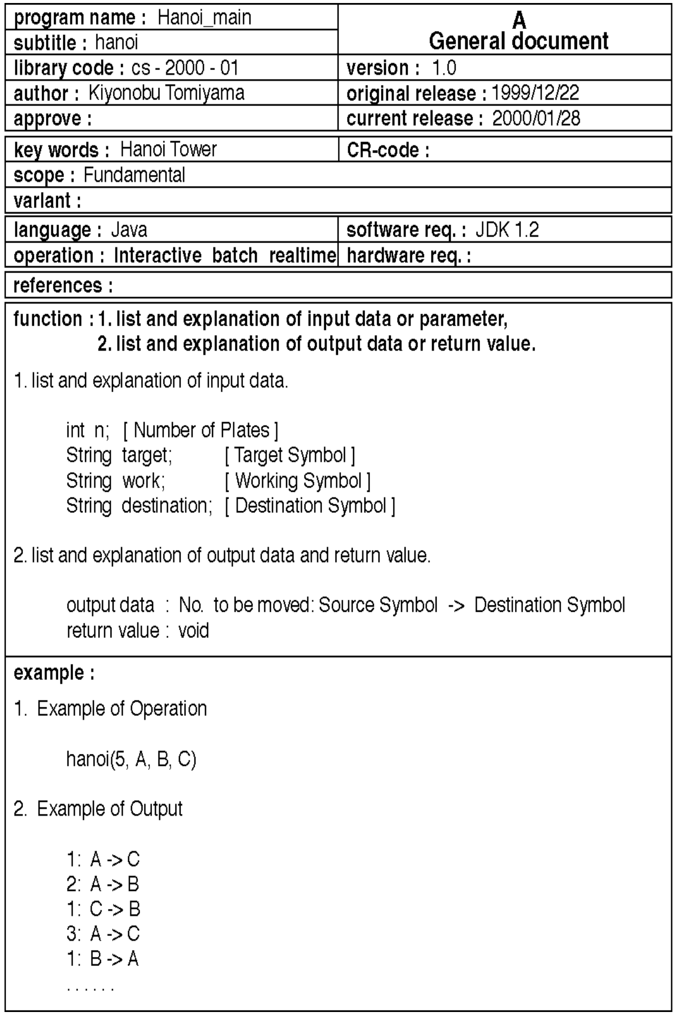
2. Preliminaries

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	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 :

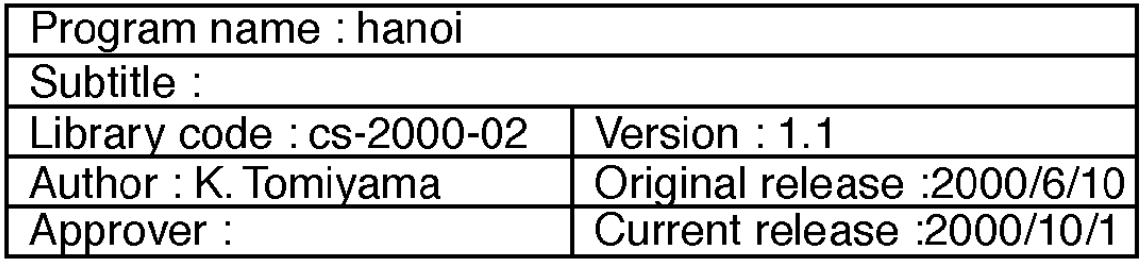
1. Example of Operation

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

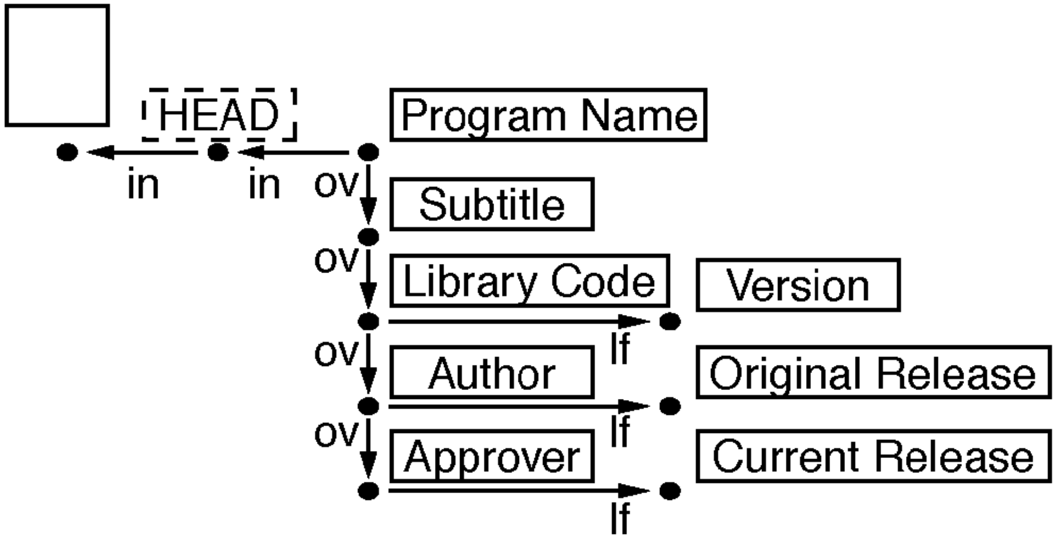
2. Example of Output

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

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

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

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

Att= $\underset{Y\in V}{\dagger}$ Att（Y） （Att(Y)=Inh(Y)∪Syn(Y)）

F= $\underset{p\in P}{\dagger}$ F_p ：Semantic rules of AGG

edNCE Graph Grammar[6]

Definition

edNCE graph grammar：G= (Σ, Δ, Γ, Ω,P,S)

Σ：The alphabet of node labels

Δ ⊆ Σ：The alphabet of terminal node labels

Γ：The alphabet of edge labels

Ω ⊆ Δ： The alphabet of final edge labels

P： The finite set of productions

S∈ Σ - Δ： The initial nonterminal

edNCE (continue)

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

$X \in \Sigma - \Delta$

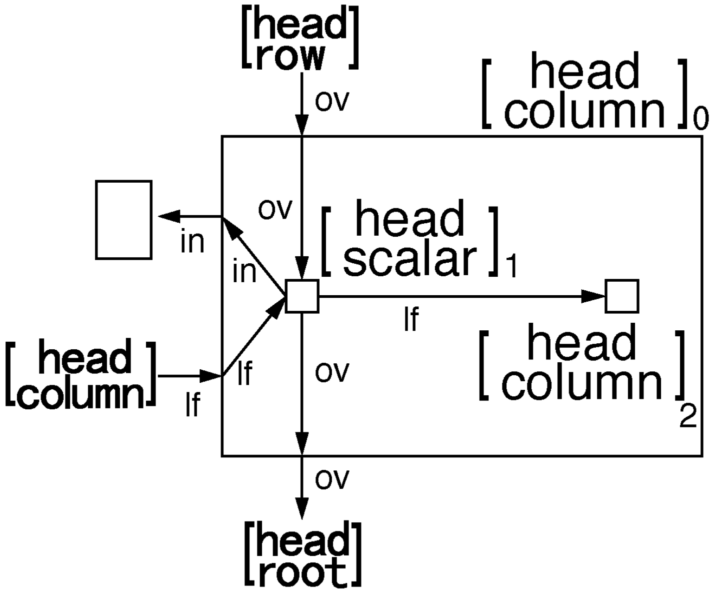
$(D,C) \in GRE_{\Sigma,\Gamma}$

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

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

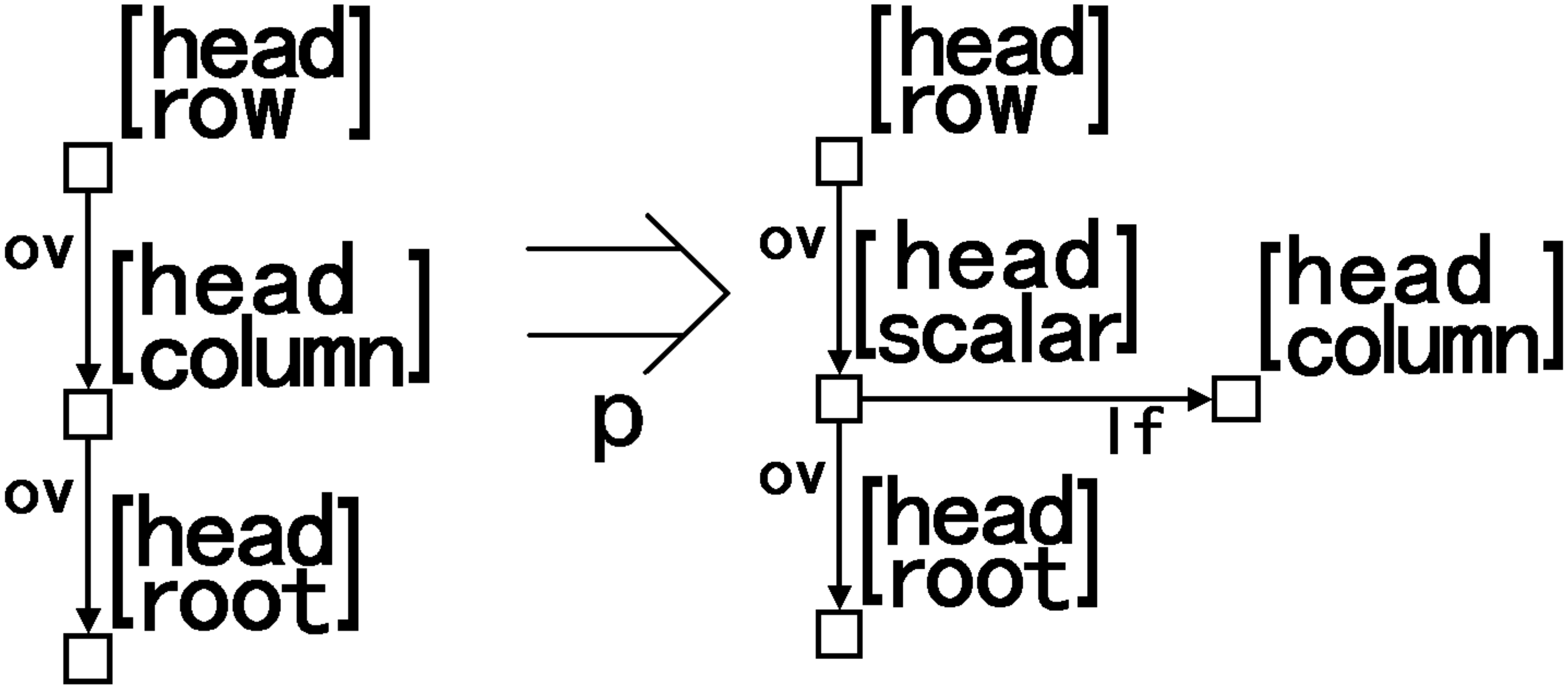
Production

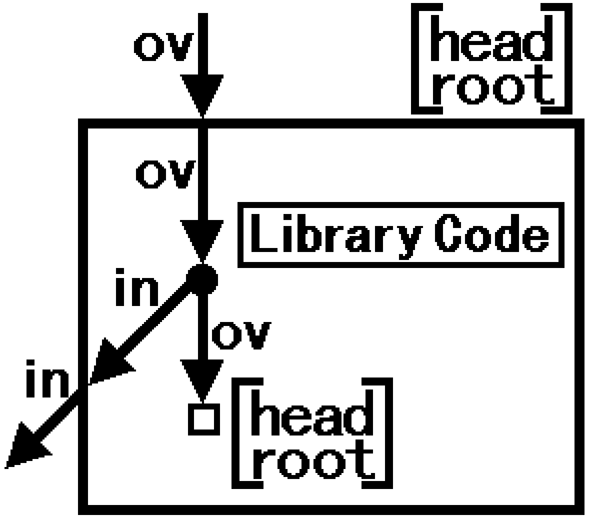
p :



C={([head-row],ov/ov,1,in),(□,in/in,1,out),
([head-column],lf/lf,1,in),([head-root],ov/ov,1,out)}

Derivation


[head row]
ov
[head column]
ov
[head root]
p
[head row]
ov
[head scalar]
lf
[head column]
ov
[head root]



C={ (α ,ov/ov,1,in) ,(β ,in/in,1,out)} such that ∀ α , β ∈ Σ

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_2exists in node labels of D_1

The composite production copy p：$X_1\rightarrow(D,C)$

is defined as follows：

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

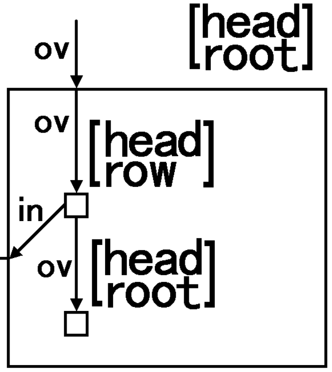
$C=\{(\sigma,\beta/\gamma,\omega,d)\in C1|\omega\in V_{D1}-V_{X2}\}$

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

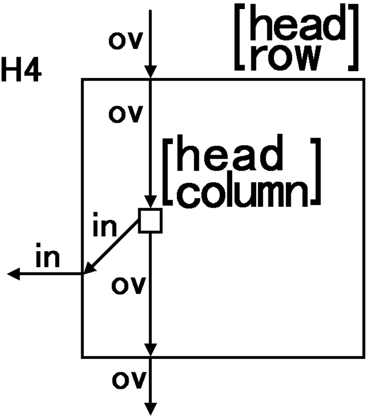
Denoted by p1○p2

COMPOSITION OF PRODUCTION COPIES

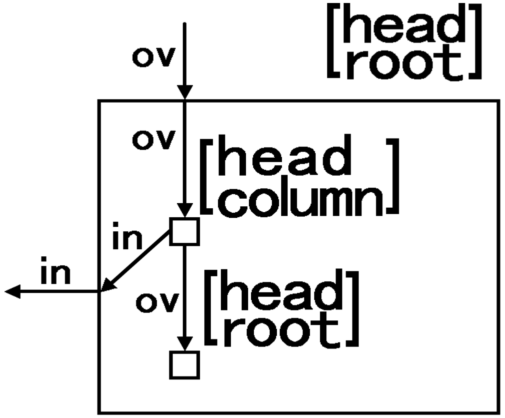
p_{H2}



p_{H4}



p_{H2} □ p_{H4}



2.3 Confluence Property [6]

Definition

An edNCE graph grammar G= (Σ,Δ,Γ,Ω,P,S) is confluent if the following holds for every sentential form H of G:

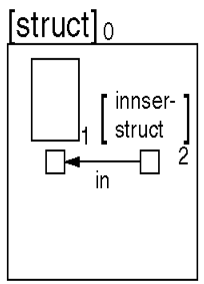
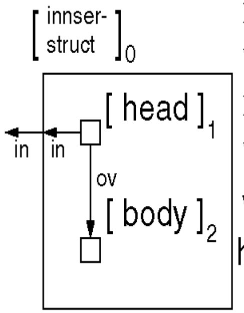
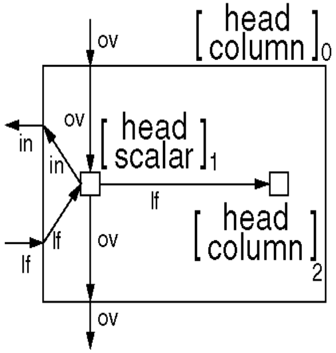
If $\boxed{H \Rightarrow_{u1,p1} H_1 \Rightarrow_{u2,p2} H_{12}}$ and $\boxed{H \Rightarrow_{u2,p2} H_1 \Rightarrow_{u1,p1} H_{21}}$ (p1,p2∈P)
are derivation of G with
u1,u2∈VH and u1≠u2,
then $\boxed{H12=H21}$

2.4 HNGG [10]

Hiform Nested Graph Grammar

HNGG=$<G_N, A_N, F_N,>$

Underlying graph grammar G_N= $(\Sigma_N, \Delta_N, \Gamma_N, \Omega_N, P_N, S_N)$ (context-free edNCE graph grammar)

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}	 ov [head column]0 ov in in [head scalar]1 lf lf lf ov [head column]2 ov	$x(1) = x(0)$ $x(2) = x(0)+width(1)+HSh$ $y(1) = y(0)$ $y(2) = y(0)$ $width(0) = width(1)+width(2)+HSh$ $height(0) = max(height(1), height(2))$

3 Editing of Nested Tabular Form

3 Editing of Nested Tabular Form

3. 1 Production Instance

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

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

$p_i \in P$:a production

H'_{p_i} :an embedded graph isomorphic H_{p_i} during $D_{i-1} \Rightarrow_{pi} D_i$

$D_{i-1} \overset{\omega H'pi}{\underset{pi}{\Rightarrow}} D_i$:

D_{i-1} is directly derived D_i by applying the (ω, p_i, H'_{p_i})

3.2 Syntactic Insertion

3.2 Syntactic Insertion

Definition (insertable)

For the derivation sequence

$D_0 \overset{\omega 1\, H'p1}{\underset{\mathbf{p}1}{\Rightarrow}} \cdots \overset{\omega i-1\, H'pi-1}{\underset{\mathbf{p}i-1}{\Rightarrow}} \boxed{D_{i-1}} \boxed{\overset{\omega\, H'pi}{\underset{\mathbf{p}i}{\Rightarrow}} D_i \overset{\omega i+1\, H'pi+1}{\underset{\mathbf{p}i+1}{\Rightarrow}} \cdots \overset{\omega n\, H'pn}{\underset{\mathbf{p}n}{\Rightarrow}} D_n}$ $(p_j : X_{p_j} \to (H_{p_j}, C_{p_j}), 1 \leqq j \leqq n)$,

Production q is <u>insertable</u> (for pi) :

Instance $(\omega_{,q}, H'_q)$ $(q = X_q \to (H_q, C_q) \in P)$ such that $\boxed{D_{\,i-1} \overset{\omega\, H'q}{\underset{q}{\Rightarrow}} Q,}$

$D_{i-1} \overset{\omega\, H'q}{\underset{q}{\Rightarrow}} Q \boxed{\overset{\omega'\, H'pi}{\underset{\mathbf{p}i}{\Rightarrow}} D'_i \overset{\omega i+1\, H'pi+1}{\underset{\mathbf{p}i+1}{\Rightarrow}} \cdots \overset{\omega n\, H'pn}{\underset{\mathbf{p}n}{\Rightarrow}} D'_n}$ exists.

□

Since a production $q \in P_N$ is insertable for p_i , then insertion of an production q into an production sequence $(p_1 , \cdots, p_i , \cdots , p_n)$ makes an production sequence

$(p_1 , \cdots, p_{i-1} , q , p_i , \cdots, p_n)$ which derives a graph D'_n as follows.

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

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

2. Apply the production q to D_{i-1}, and get the resultant graph Q .

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

3. Apply the production sequence $(p_i , p_{i+1} , \bullet \bullet \bullet, p_n)$ to Q, and get the resultant graph D'_n.

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

Definition

To insert a source graph *A* at edge *x* in a target graph *H*.

$$\overset{def}{\Leftrightarrow}$$

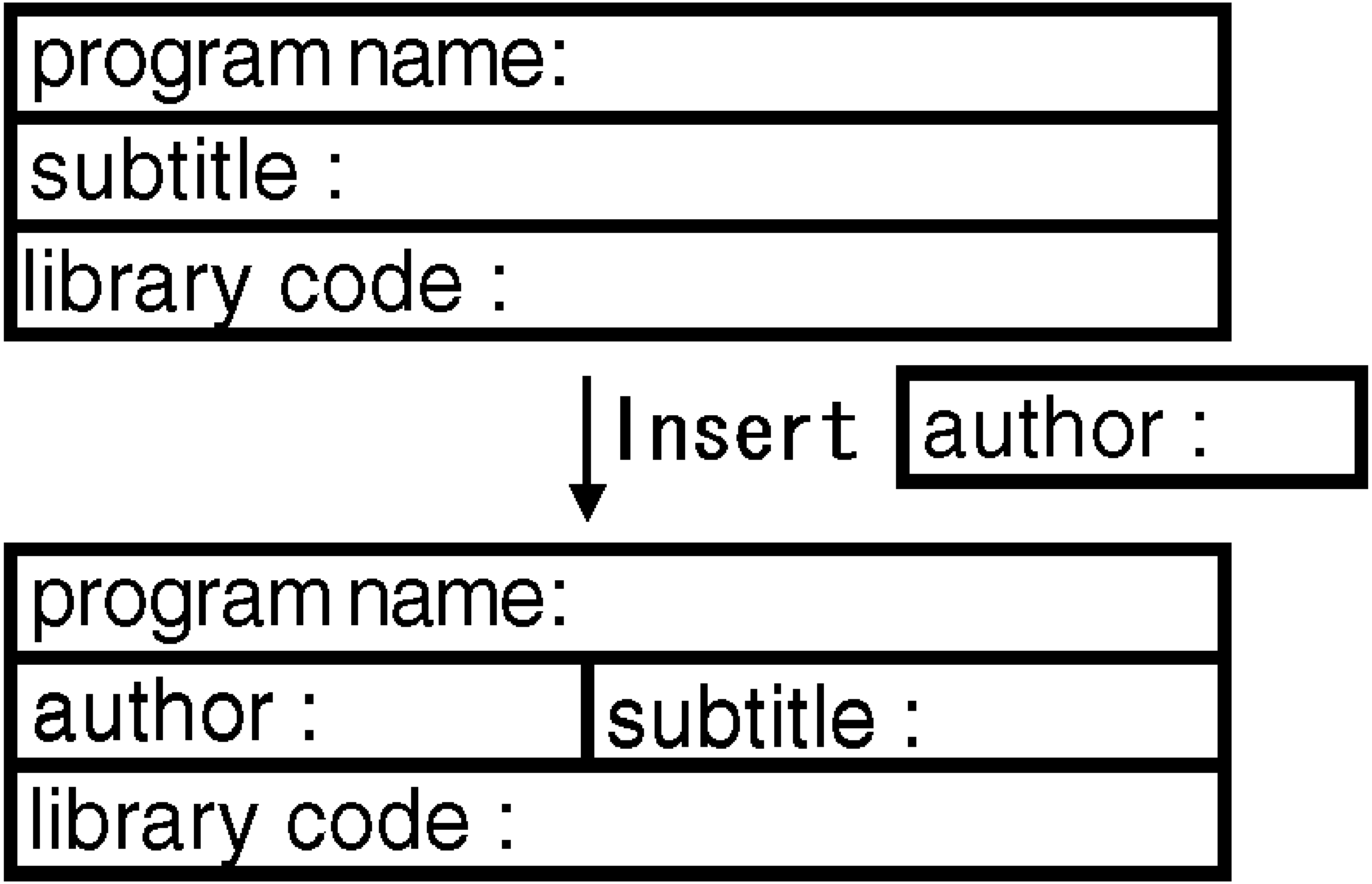
1. A composite production copy *q* for the graph *A* exists.

2. The composite production copy *q* can be insertable at the edge *x* in the graph *H*.

3. $H \rightarrow \bullet\bullet\bullet \longrightarrow H'$ **:**

***H'* is the inserted graph which inserts the graph *A* at the edge *x*.**

□



program name:

subtitle :

library code :

Insert

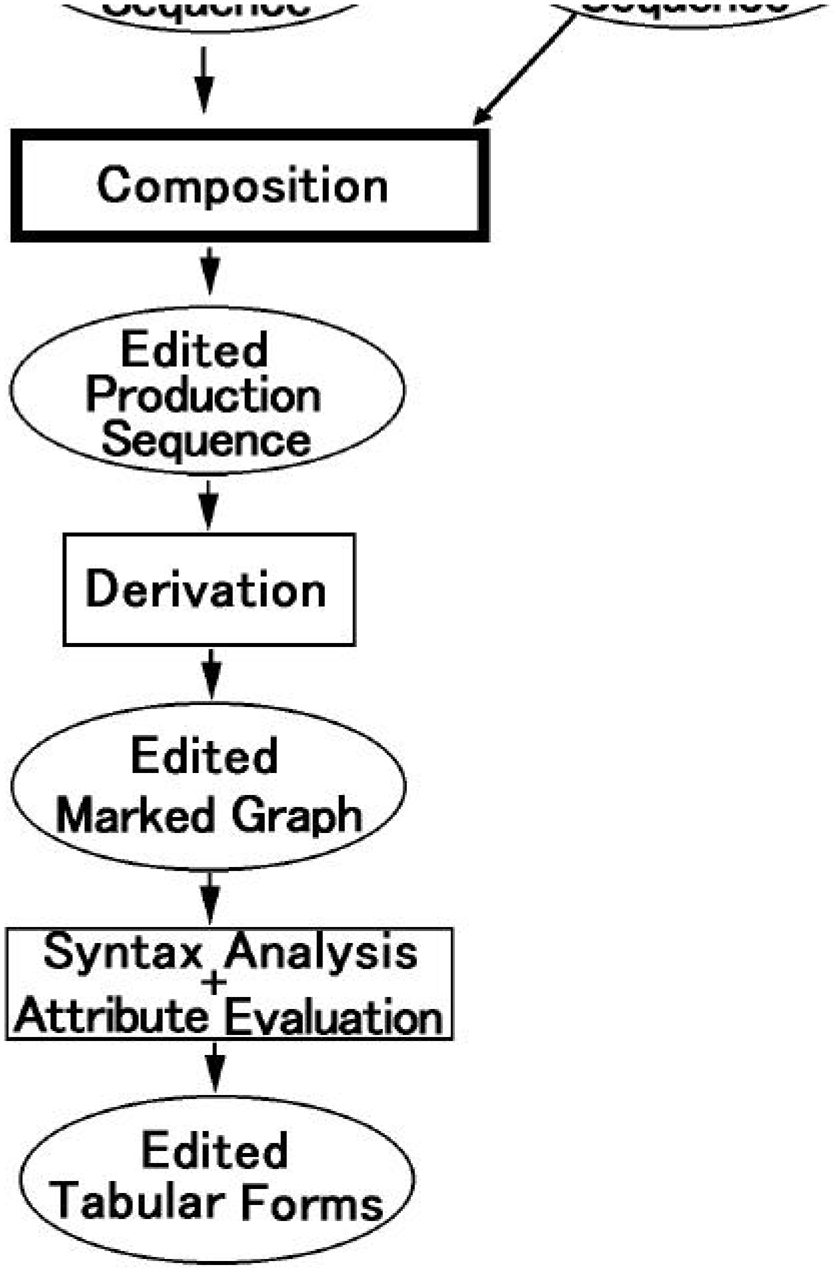
author :

program name:

author :

subtitle :

library code :



A process flow for an insertion of Hiform editing system

3.3 Syntactic Deletion of Item

3.3 Syntactic Deletion of Item

Definition (deletable)

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

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

Node u is not applied to any production after that.

Production $p=X_p \rightarrow (D_p,C_p) \in P_N$ is deletable if one of the following Assumptions 1-3 is met

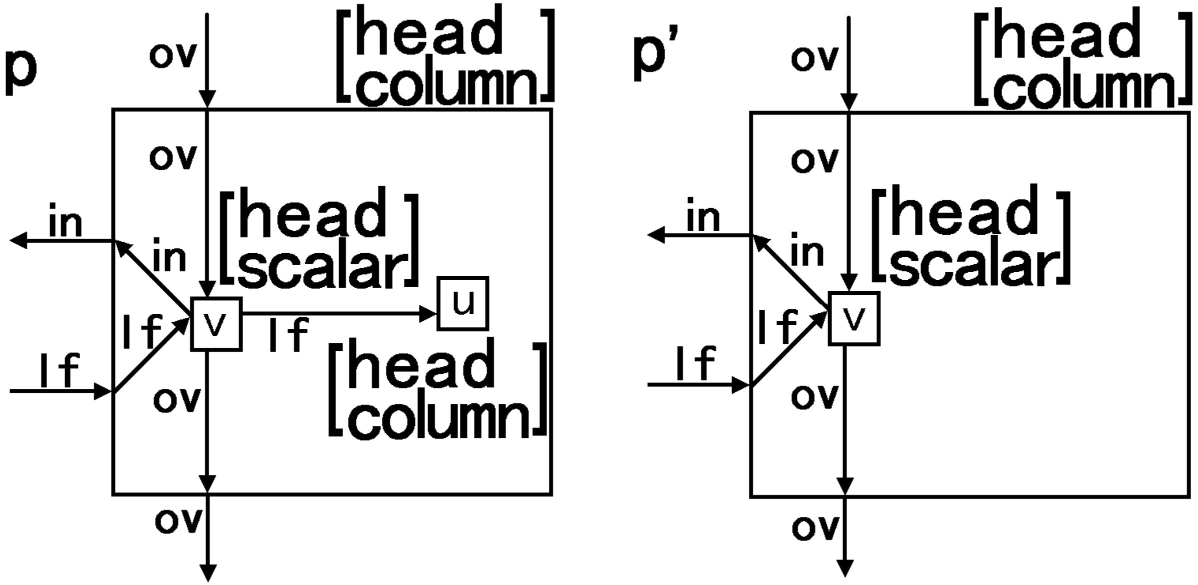
Assumption 1

For $p \in P_N$, $\exists$ p': $X_p \to D({}_p, C_p) \in P_N$ s.t.

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

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

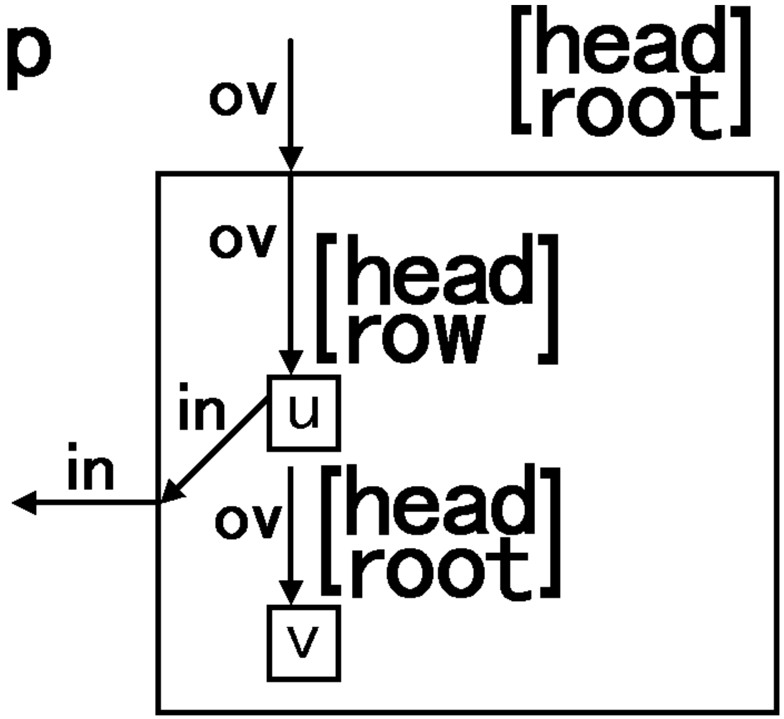
For example:



Assumption 2

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

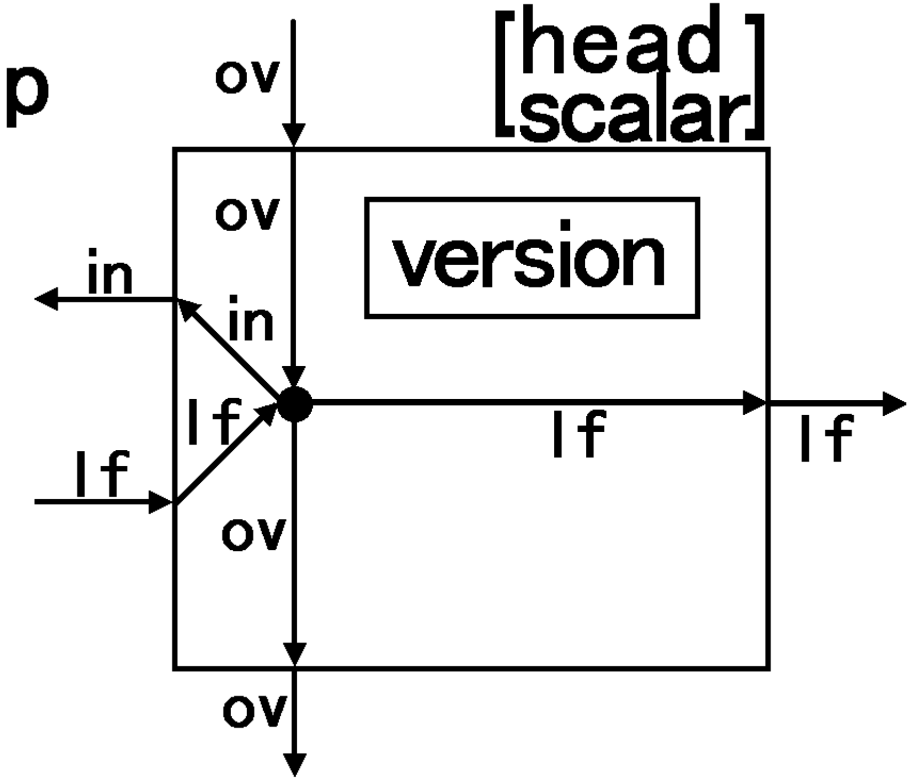
For example:



Assumption 3

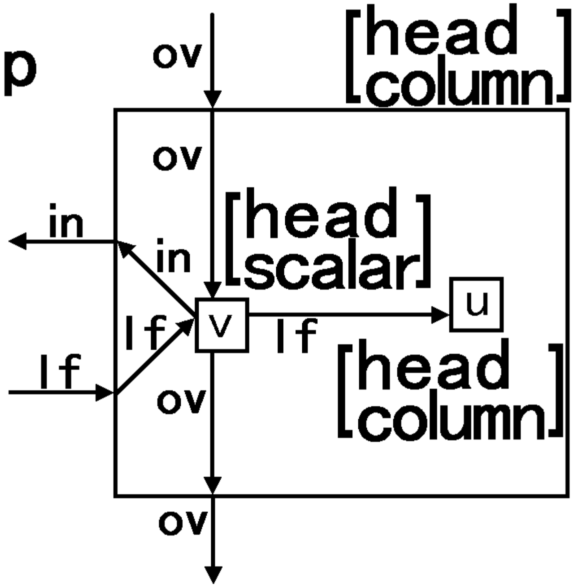
$$\omega_j \notin H'_p \; , \quad 1 \leqq j \leqq n$$

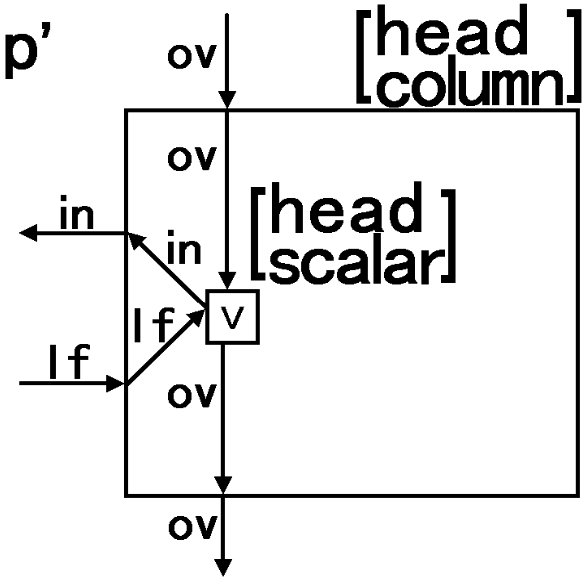
For example:



□

The case of Assumption1





Since a production $p \in P_N$ is deletable, deletion of a production p from a production sequence

$L=(p_1 ,\cdots,p_k ,$ p , $p_l ,\cdots, p_n)$ makes an production sequence $(p_1,\cdots, p_k ,$p'$, p_l , \cdots , p_n)$,which derives a graph D'n as follows.

1. Trace the derivation sequence Dn back to F.

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

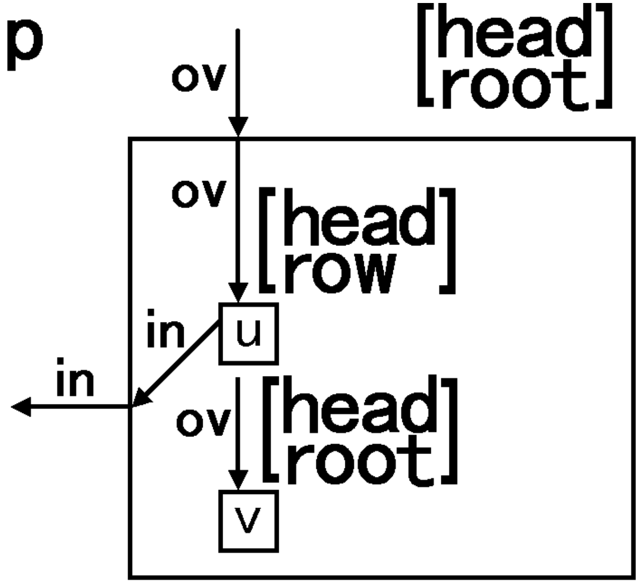
2. Apply the production p’ to F, and get the resultant graph D'p.

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

3. Apply the production sequence (p_l , • • •, p_n) to D'p, and get the resultant graph D’n.

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

The case of Assumption 2



Since a production $p \in P_N$ is deletable, deletion of a production p from a production sequence

$L=(p_1, \cdots, p_k, p, p_l, \cdots, p_n)$ makes an instance sequence $(p_1, \cdots, p_k, p_l, \cdots, p_n)$, which derives a graph D‘n as follows.

1. Trace the derivation sequence Dn back to F.

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

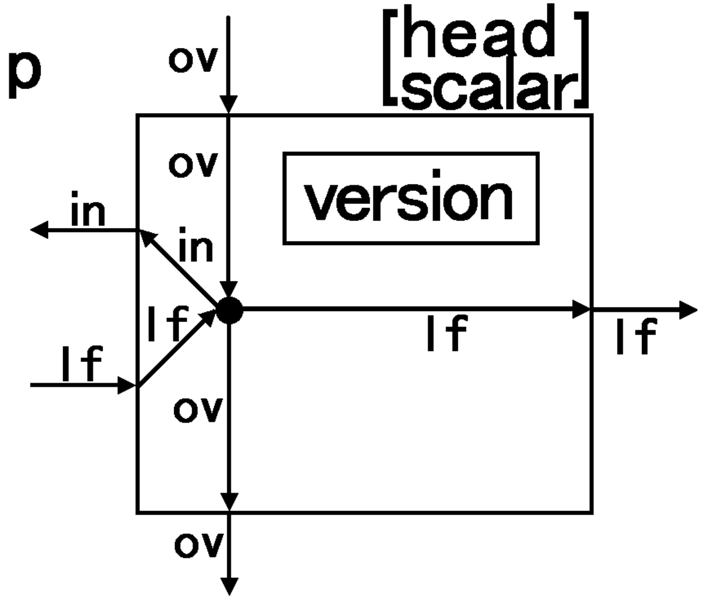
2. Rename the node $\omega \in V_F$ as v, and get the resultant graph F'.

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

3. Apply the production sequence $(p_l$, ..., $p_n)$ to F', and get the resultant graph D'n.

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

The case of Assumption 3



Since a production $p \in P_N$ is deletable, deletion of a production p from an instance sequence $L=(p_1,\bullet\bullet\bullet,p_k\ ,\ p\ ,p_l\ ,\bullet\bullet\bullet,p_n)$makes an instance sequence $(p_1,\bullet\bullet\bullet p_k\ ,\ p_l\ ,\ \bullet\bullet\bullet,p_n\ )$, which derives a graph D’n as follows.

1. Trace the derivation sequence Dn back to F

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

2. Apply the production sequence $(p_l, \ldots, p_n)$ to F, and get the resultant graph D’n.

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

Definition

To <u>delete</u> a node *A* from a graph *H*

$\overset{def}{\Leftrightarrow}$

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

②The production *q* is deletable in the instance sequence for graph *H*.

③ *H*→ ••→*H* ’:

H' is the deleted graph which deletes the node *A* in the graph *H*.

3.4 Deletion of Blocked Items

Let D=(V_D,E_D,λ_D) be a graph.Let T⊆D be a subgraph.

Then, the deletion of the production about the derivation of T has been performed as follows.

1. D ’＝D
2. Let T⊆D' be a subgraph.
3. In derivation sequence D0⇒・・⇒D',

   q can be removed from the production in T.

   a) If q exists, the graph which removed q from the sequence, and then renew D' and return to 2.

   b) It is finished if q does not exist.

3.5 Property of Editing Method

Theorem 3.1

Deletion (insertion, block deletion) in HNGG is executed in linear time.

□

Theorem 3.2

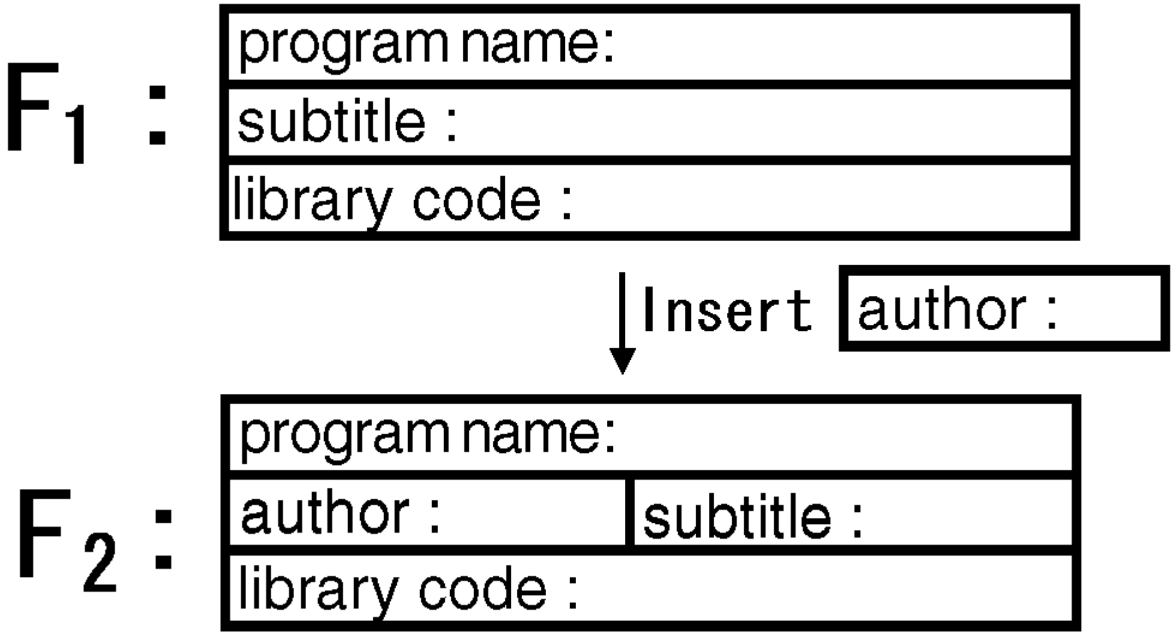
Let H be the graph obtained from G by the deletion of nodes a and b in this order, in HNGG.
Let H' be the graph obtained from G by the deletion of nodes b and a in this order, in HNGG.
Then, H=H'.

□

4 Example： Insertion Process

4 Example: Insertion Process

Insertion of author : in the left side of 'subtitle' in form F_1



F_1 :

program name:

subtitle :

library code :

Insert author :

F_2 :

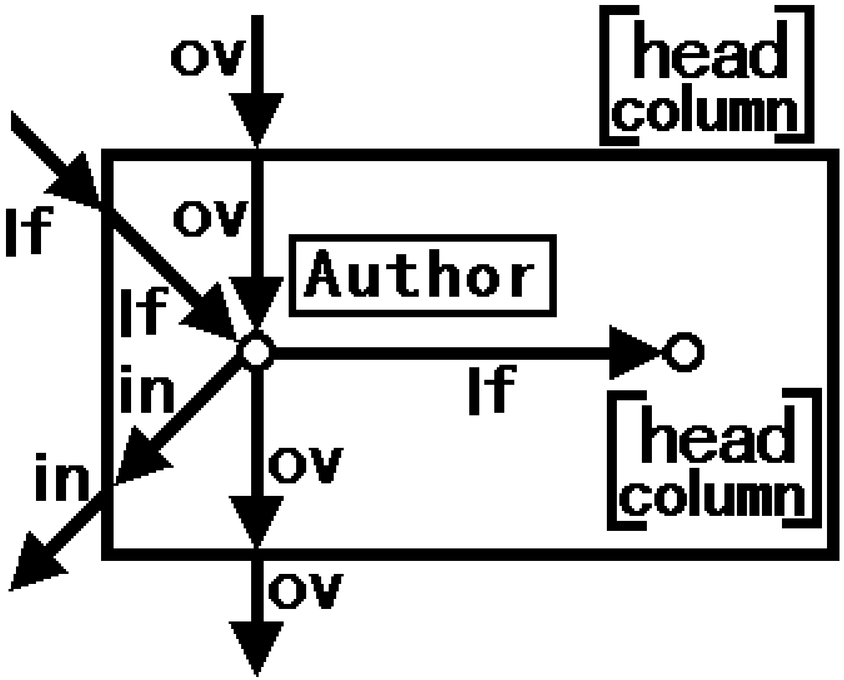
program name:

author : subtitle :

library code :

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

$P_{Q} = P_{H5} \circ P_{H11}$

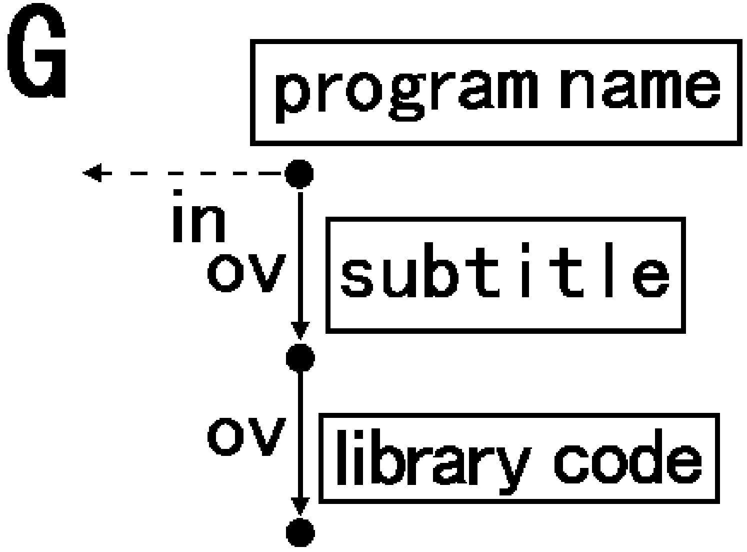


Step2. Construction of the Derivation D_1 of graph G for form F_1.

$$D_1 : G_0 \xRightarrow[\mathbf{p}1]{\omega 1\ \mathrm{H'p1}} G_1 \xRightarrow[\mathbf{p}2]{\omega 2\ \mathrm{H'p2}} G_2 \xRightarrow[\mathbf{p}H1]{\omega H1\ \mathrm{H'pH1}} G_3 \xRightarrow[\mathbf{p}H2]{\omega H2\ \mathrm{H'pH2}} G_4 \xRightarrow[\mathbf{p}H4]{\omega H4\ \mathrm{H'pH4}} G_5 \xRightarrow[\mathbf{p}H6]{\omega H6\ \mathrm{H'pH6}} G_6 \xRightarrow[\mathbf{p}H7]{\omega H7\ \mathrm{H'pH7}}$$

$$G_7 \xRightarrow[\mathbf{p}H2]{\omega H2\ \mathrm{H'pH2}} G_8 \xRightarrow[\mathbf{p}H4]{\omega H4\ \mathrm{H'pH4}} G_9 \xRightarrow[\mathbf{p}H6]{\omega H6\ \mathrm{H'pH6}} G_{10} \xRightarrow[\mathbf{p}H8]{\omega H8\ \mathrm{H'pH8}} G_{11} \xRightarrow[\mathbf{p}H3]{\omega H3\ \mathrm{H'pH3}} G_{12} \xRightarrow[\mathbf{p}H4]{\omega H4\ \mathrm{H'pH4}} G_{13}$$

$$\xRightarrow[\mathbf{p}H6]{\omega H6\ \mathrm{H'pH6}} G14 \xRightarrow[\mathbf{p}H9]{\omega H9\ \mathrm{H'pH9}} G$$

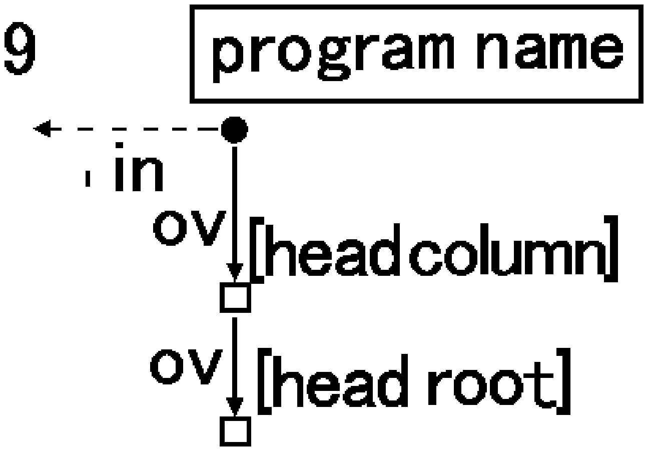


Step3. Find a target graph G_9 by Insertion Point in G Find a subderivation D_{11} to generate G_9 in D_1.

$$D_{11} : G_0 \overset{\omega 1\ H'p1}{\underset{\mathbf{p}1}{\Rightarrow}} G_1 \overset{\omega 2\ H'p2}{\underset{\mathbf{p}2}{\Rightarrow}} G_2 \overset{\omega H1\ H'pH1}{\underset{\mathbf{p}H1}{\Rightarrow}} G_3 \overset{\omega H2\ H'pH2}{\underset{\mathbf{p}H2}{\Rightarrow}} G_4 \overset{\omega H4\ H'pH4}{\underset{\mathbf{p}H4}{\Rightarrow}} G_5 \overset{\omega H6\ H'pH6}{\underset{\mathbf{p}H6}{\Rightarrow}} G_6$$

$$\overset{\omega H7\ H'pH7}{\underset{\mathbf{p}H7}{\Rightarrow}} G_7 \overset{\omega H2\ H'pH2}{\underset{\mathbf{p}H2}{\Rightarrow}} G_8 \overset{\omega H4\ H'pH4}{\underset{\mathbf{p}H4}{\Rightarrow}} G_9$$

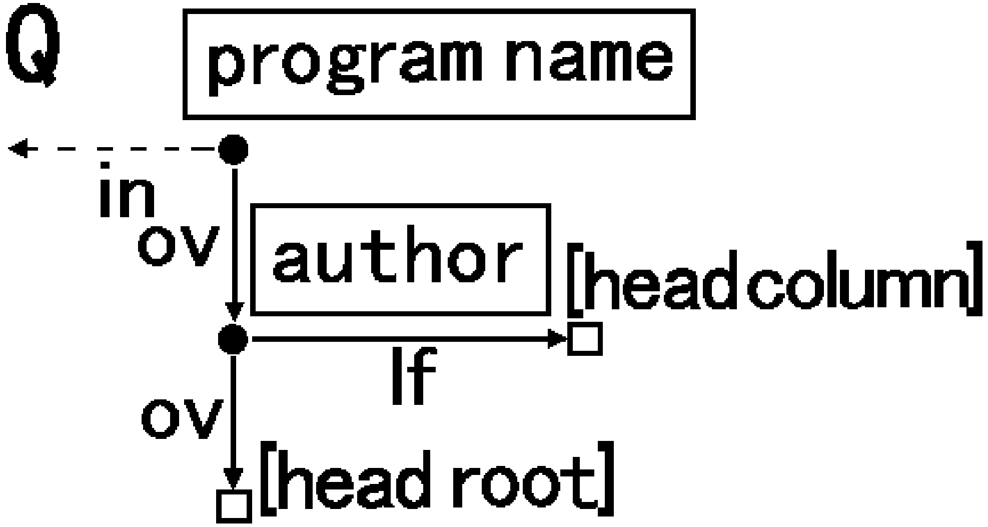
G_9



Step4. Apply P_Q to G_9 and obtain G_Q.

$$G_9 \overset{\omega QH'pQ}{\underset{\mathbf{p}Q}{\Rightarrow}} G_Q$$

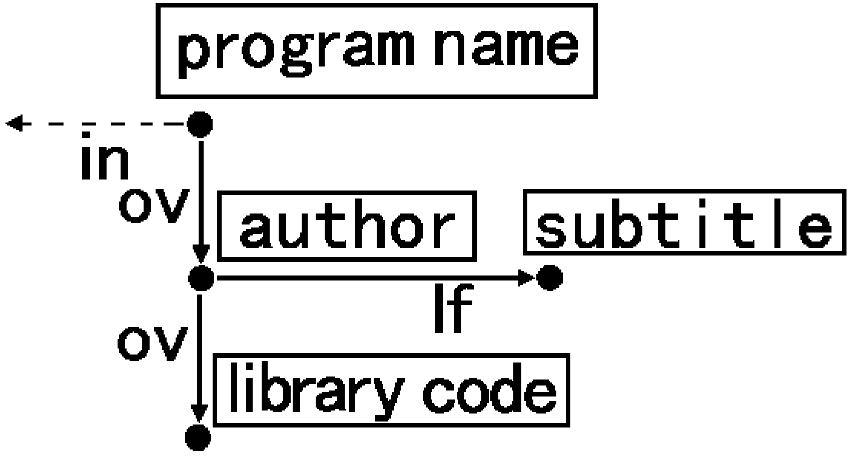
G_Q



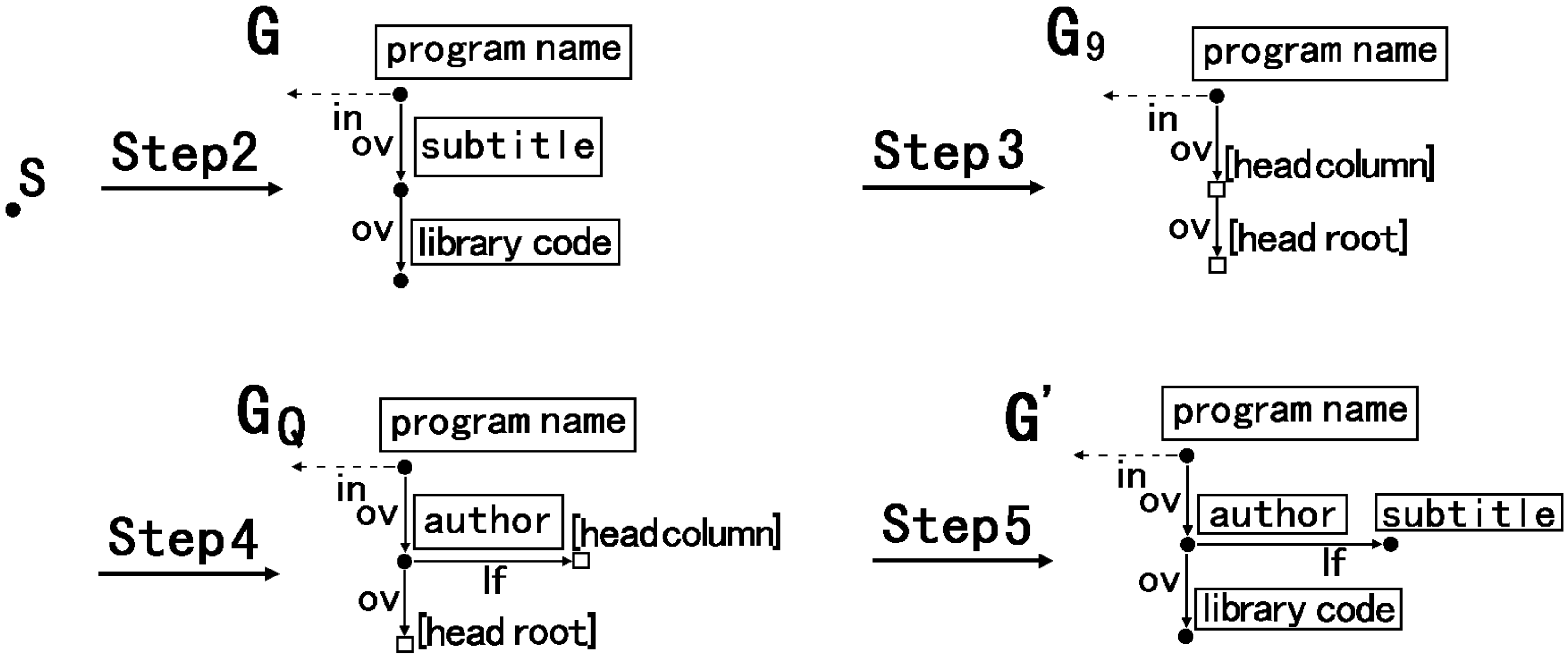
Step5. Apply the latter part of D_1 to G_Q.

$$G'_Q \overset{\omega H6\,\mathrm{H'pH6}}{\underset{\mathbf{p}H6}{\Rightarrow}} G'_{10} \overset{\omega H8\,\mathrm{H'pH8}}{\underset{\mathbf{p}H8}{\Rightarrow}} G'_{11} \overset{\omega H3\,\mathrm{H'pH3}}{\underset{\mathbf{p}H3}{\Rightarrow}} G'_{12} \overset{\omega H4\,\mathrm{H'pH4}}{\underset{\mathbf{p}H4}{\Rightarrow}} G'_{13} \overset{\omega H6\,\mathrm{H'pH6}}{\underset{\mathbf{p}H6}{\Rightarrow}} G'_{14} \overset{\omega H9\,\mathrm{H'pH9}}{\underset{\mathbf{p}H9}{\Rightarrow}} G'$$

G'



G' is a graph for form F_2.



Insertion process

6.Conclusion

- We proposed editing methods for tabular forms, based on the attribute edNCE graph grammar.
- Examples to apply editor methods were shown.

Future works.

- Detailed algorithm of editing methods.
- Other edit manipulations representing a division manipulation, a combination manipulation and so on.
- We are now developing a tabular form editor system utilizing this approach.