

# 順位グラフ文法によるプログラム仕様書の定式化

有田友和 (日本大学)

杉田公生 (東海大学)

○松本亮 (日本大学)

第4回グラフ書き換え系と  
応用ワークショップ

1999年2月4日

背景

ソフトウェアプロセスの記述法と  
支援システムは基本的問題



ソフトウェア文書  
国際規格ISO6592(1985)



仕様書の管理文書としての  
重要性の高まり



プログラム仕様書  
Hiform96(定型用紙)  
(杉田他,1997)

順位グラフ文法  
(Franck,1978)

⇓ 拡張

⇓ 応用

★今回、仕様書をグラフ文法により定式化

## 従来の仕様書の問題点

### 定型用紙

項目の数，サイズ，順番を固定⇒自由度が少ない

### 自由書式

項目の数，サイズ，順番を自由⇒自由度が高い



- ・描画が困難
- ・正しくない仕様書が出来やすい

## 研究目的

### 仕様書をグラフ文法による定式化



項目の数，サイズ，順番を少し自由にしたうえで

- ★正しい仕様書が厳密に定まる
- ★描画が正確に行える

## 結果

- Hifrom96 様式 A1 に関し生成規則および順位関係表を作成

	A1 共通部分	A1 独自部分	合計
生成規則数	1 6	1 8	3 4
順位関係数	7 3 5	8 6 7	1 6 0 2

- 生成規則で A1 が生成可能
- 生成規則 , 順位関係表により A1 が構文解析可能

## 2. 準備

### 2.1 仕様書 Hiform96[6,7,8]

- JISX0126-1987のガイドライン項目をすべて含む
- 記述形式は表形式
  - ★ 関連項目を A4サイズ1P に配置 (様式)
  - ★ 様式は17種類
  - ★ 作成・管理等の目的に適している
- 各様式の並べ方は句構造文法で定義
  - ★ 様式内項目の 配置は固定

# 例1 . Hiform96の様式A1

プロジェクトコード:	A
プログラム名:	プログラム概要書
ライブラリ登録コード:	版名:
著作者:	初版発行日:
文書責任者:	現行版発行日:
キーワード:	CR分類コード:
目的・範囲:	
背景情報:	
記述言語:	所要ソフトウェア:
操作:	所要ハードウェア:
関連文書:	
機能:	
例:	

## 定義1 [1]

マーク付きグラフ (marked graph) は

$G = (K, R, k, r)$  の4項組

$K$  : 頂点の有限集合 ( $K \neq \phi$ )

$R$  : 有向辺の有限集合 ( $\subset K \times K$ )

$V$  : 頂点のマークの有限集合

$M$  : 辺のラベルの有限集合

$k : K \rightarrow V$  (頂点をマーク付けする写像)

$r : R \rightarrow M$  (辺をラベル付けする写像)

□

## 定義2 [1]

文脈自由生成規則 (context-free production) は

$p = (A, H, p^e, p^s)$  の4項組

$A$  : 一つの頂点のグラフ ( $p$  の左辺)

$H = (K_h, R_h, k_h, r_h)$  : 空でないグラフ ( $p$  の右辺)

$p^e, p^s : M \rightarrow K_h$

( $M$  から  $K_h$  への部分関数)

□

グラフ  $G = (K, R, k, r)$ ,  $G' = (K', R', k', r')$  ,

$p = (A, H, p^e, p^s)$   $H = (K_h, R_h, k_h, r_h)$

に対し関係  $G \xrightarrow[p]{\Rightarrow} G' \stackrel{\text{def}}{\Leftrightarrow}$

1.  $k(i) = A$  であるような  $i \in K$  が存在する

2.  $K' = (K - \{i\}) \cup R_h$

3.  $k'(a) = \begin{cases} a \in K \text{ に対し } k(a) \\ a \in K_h \text{ に対し } k'(a) \end{cases}$

4.  $m \in M$  に対し,

$R' = R - \{(a, b) \mid a = i \text{ 又は } b = i\} \cup R_h$

$\cup \{(a, x) \mid \bigvee_{(a,i) \in R} r((a, i)) = m \text{ かつ } p^e(m) = x \in K_h\}$

$\cup \{(x, b) \mid \bigvee_{(i,b) \in R} r((i, b)) = m \text{ かつ } p^s(m) = x \in K_h\}$

5.  $r'((a, b)) = \begin{cases} (a, b) \in R \text{ に対し } r((a, b)) \\ (a, b) \in R_h \text{ に対し } r_h((a, b)) \\ R' \text{ で新たに追加された辺については} \\ G \text{ の辺のラベルを継承する} \end{cases}$

□



**定義3**[1]

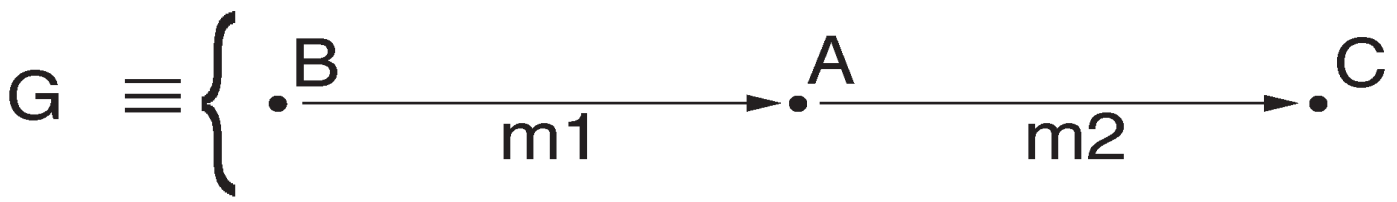
$G, G'$ をマーク付きグラフとする

$G'$ は $G$ から 導出(derive)される  $\stackrel{\text{def}}{\iff}$

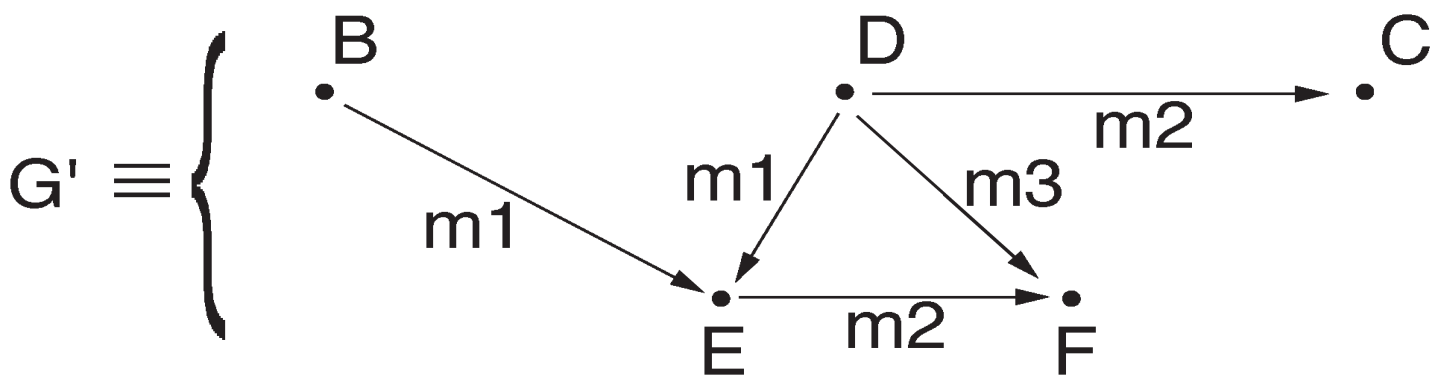
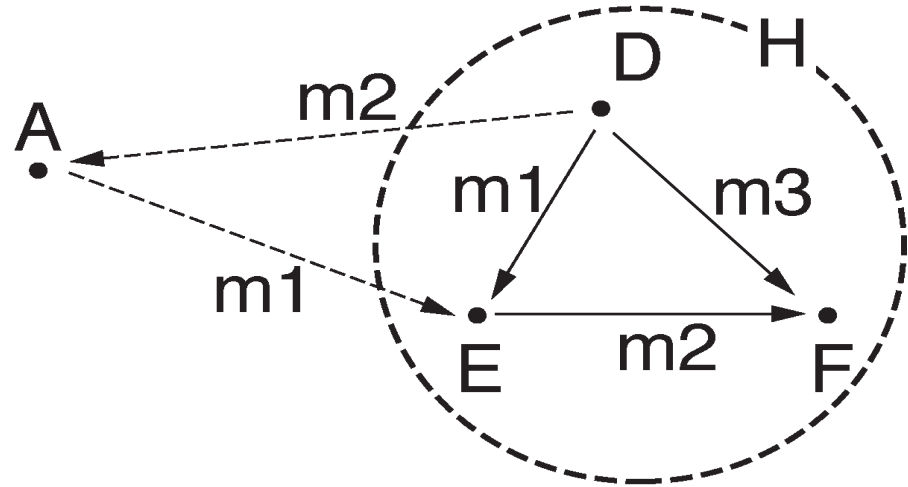
ある生成規則  $p$  に対し  $G \xrightarrow{p} G'$

□

例2. 導出



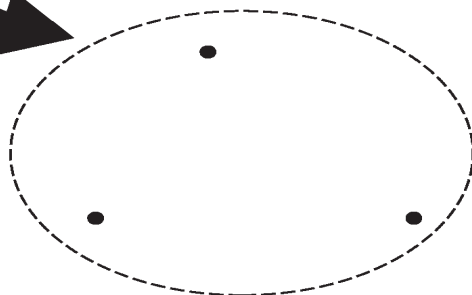
生成規則 P



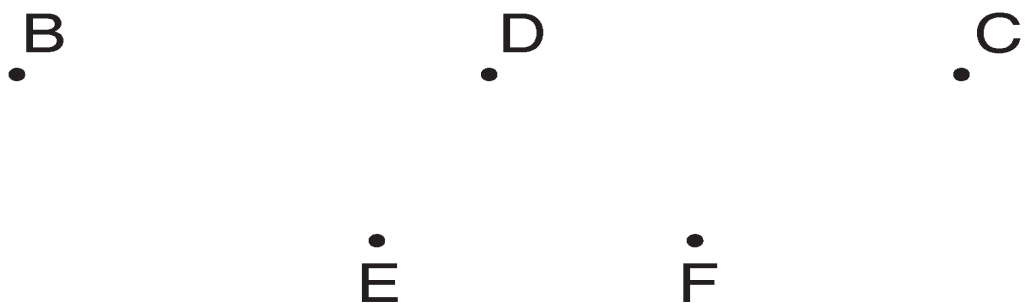
# 導出メカニズム



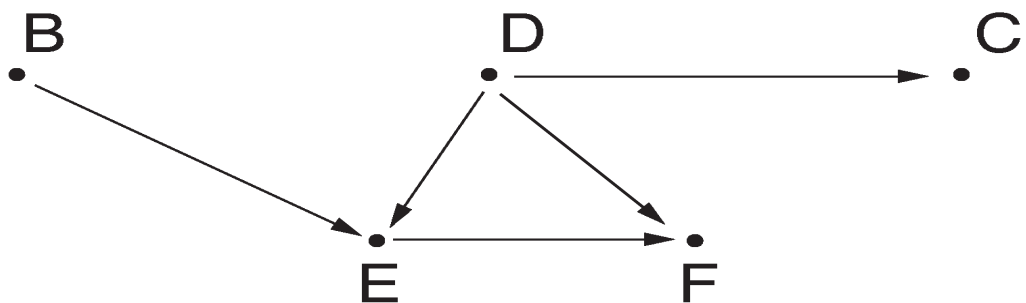
(2)



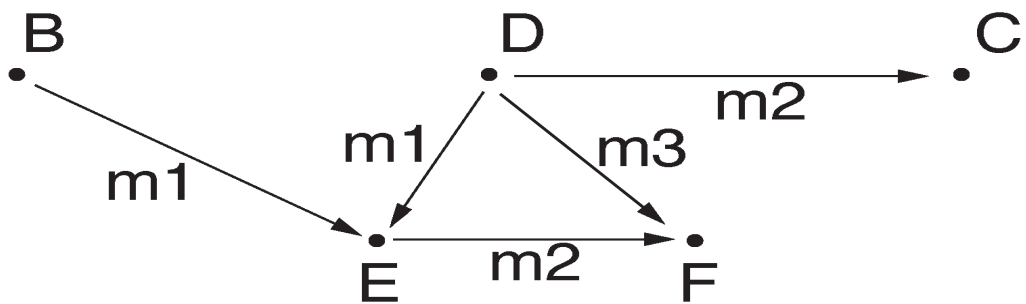
(3)



(4)



(5)



#### 定義4 [1]

文脈自由グラフ文法 (a context-free graph grammar) は  $GG = (V, T, M, P, S)$  の5項組

$V$  : 頂点のマークの有限集合

$T$  : 終端記号 ( $\subset V$ )

$M$  : 辺のラベルの有限集合

$P$  : 生成規則  $p = (A, H, p^e, p^s)$  の有限集合

$S$  : 開始記号 ( $\in V - T$ )

□

#### 定義5 [1]

グラフ文法  $GG = (V, T, M, P, S)$ , グラフ  $G$  に対して

1.  $p \in P$  に対する関係  $\xrightarrow{p}$  の反射推移閉包を  $\xrightarrow{GG}$

2.  $S \xrightarrow{GG} G \stackrel{\text{def}}{\iff} G$  は  $GG$  によって導出される

3.  $G$  が  $GG$  の グラフ言語 に属する  $\stackrel{\text{def}}{\iff}$

$G$  が  $GG$  によって導出されかつ  $G$  のすべての頂点が終端記号によりマーク付けられている

## 定義6 [1]

グラフ文法  $G$  の 構文木 (parse tree) は以下を満たす木である

(i) 根は開始グラフに対する生成規則によりラベル付けされる

(ii) 頂点  $1, 2, \dots, n$  のラベルがちょうど生成規則  $p$  の右辺の非終端記号  $A_1, A_2, \dots, A_n$  になっている  $\Leftrightarrow$  部分写像  $f : \{1, 2, \dots, n\} \rightarrow \{s_1, s_2, \dots, s_n\}$  が存在ただし,  $s_i$  は  $A_i$  を左辺に持つ生成規則

□

## 定義8 [1]

グラフ文法は 曖昧でない (unambiguous)  $\stackrel{\text{def}}{\Leftrightarrow}$  導出可能なグラフ  $G$  に対し、 $G$  を産出する構文木がただ一つ存在

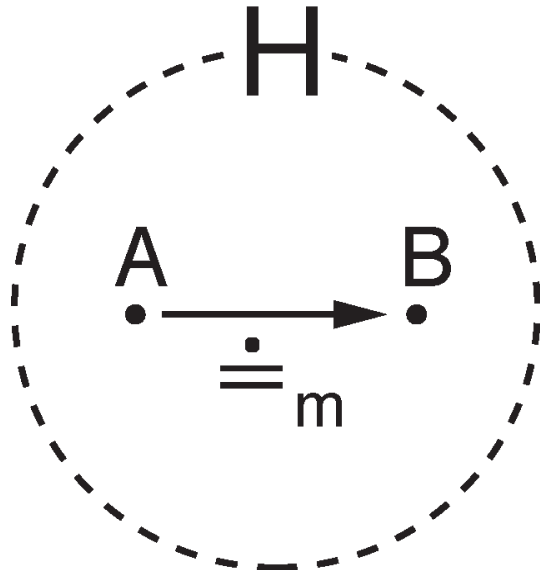
□

記法9 [1]

$m \in M$  に対し、

$\stackrel{\cdot}{=}m \stackrel{\text{def}}{=} \left\{ (A, B) \mid \begin{array}{l} A, B \in V \text{ かつ} \\ \text{辺 } (x, y) \text{ を持つ規則の右辺が存在} \\ \text{ただし } x, y \text{ はそれぞれ } A, B \text{ により} \\ \text{マーク付けされ } (x, y) \text{ はラベル } m \end{array} \right.$

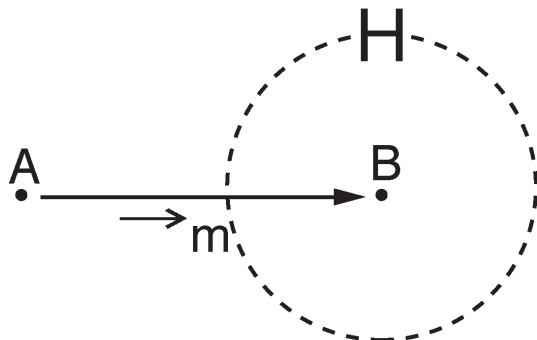
□



$m \in M$  に対し、

$$\rightarrow_m \stackrel{\text{def}}{=} \left\{ \begin{array}{l} A, B \in V \text{ かつ} \\ (A, B) \mid \text{生成規則 } p = (A, H, p^e, p^s) \text{ が存在} \\ \text{かつ } B \text{ は } H \text{ の頂点 } p^e(m) \text{ のラベル} \end{array} \right.$$

□

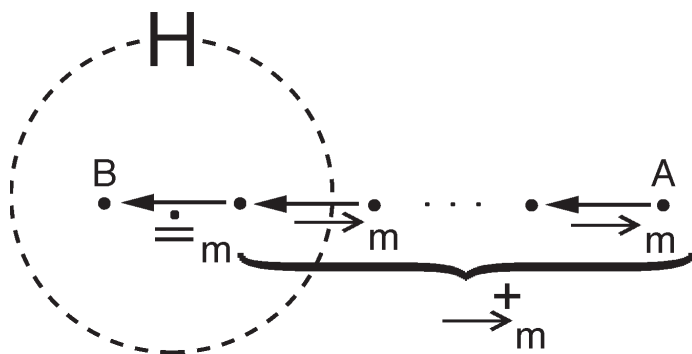


**記法 11** [1]

$m \in M$  に対し、

$$\langle \cdot \rangle_m \stackrel{\text{def}}{=} \dot{=} \langle \cdot \rangle_m \cdot \rightarrow_m^+ \text{ ただし } + \text{ は推移的閉包を表す}$$

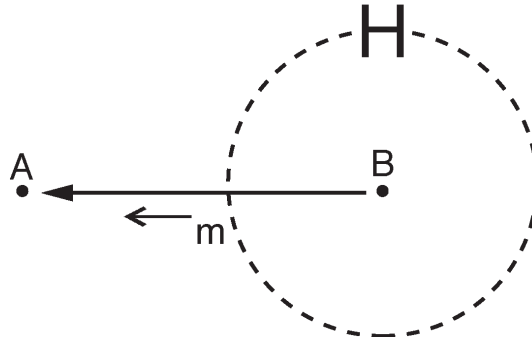
□



$m \in M$  に対し、

$$\leftarrow_m \stackrel{\text{def}}{=} \begin{cases} A, B \in V \text{ かつ} \\ (B, A) \mid \text{生成規則 } p = (A, H, p^e, p^s) \text{ が存在} \\ \text{かつ } B \text{ は } H \text{ の頂点 } p^s(m) \text{ のラベル} \end{cases}$$

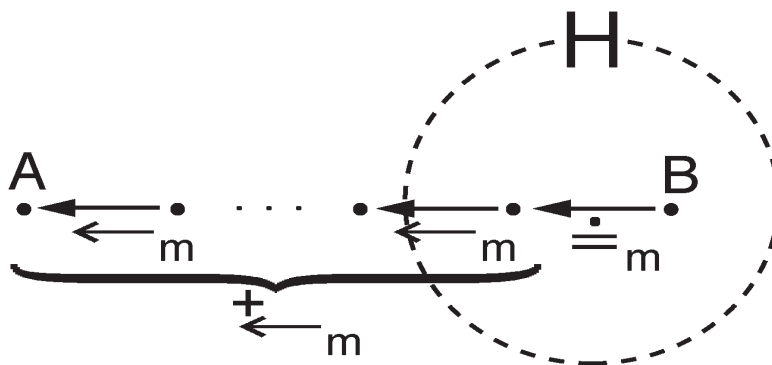
□



**記法 13**[1]

$$m \in M \text{ に対し、 } \cdot >_m \stackrel{\text{def}}{=} \leftarrow_m \cdot \stackrel{\cdot}{=} m$$

□

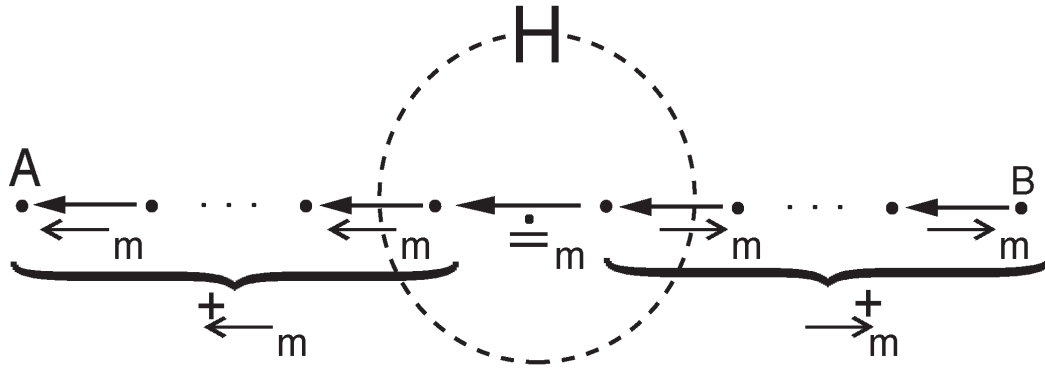




$m \in M$  に対し、

$$\langle \cdot \rangle \stackrel{\text{def}}{=} \overset{+}{\leftarrow} m \cdot \overset{\cdot}{=} m \cdot \overset{+}{\rightarrow} m$$

□



**定義 15**[1]

グラフ文法が 矛盾しない (conflictless)  $\stackrel{\text{def}}{\iff}$

すべての  $m \in M$  に対して

関係  $\langle \cdot \rangle_m, \overset{\cdot}{=} m, \overset{+}{\leftarrow} m, \overset{+}{\rightarrow} m$  が互いに素である

□

### 定義 16 [1]

順位関係 (precedence relation) は以下のように定義できる

$$\begin{aligned} < \cdot &\stackrel{\text{def}}{=} \bigcup_{m \in M} < \cdot m & \cdot > &\stackrel{\text{def}}{=} \bigcup_{m \in M} \cdot > m \\ < \cdot > &\stackrel{\text{def}}{=} \bigcup_{m \in M} < \cdot > m & &\stackrel{\text{def}}{=} \bigcup_{m \in M} \doteq m \end{aligned}$$

□

### 定義 17 [1]

グラフ文法が 順位グラフ文法 (a precedence graph grammar) である  $\stackrel{\text{def}}{\Leftrightarrow}$

- (i) 順位関係に矛盾がない
- (ii) 各還元が一意に定まる
- (iii) 文法において、再帰的な非終端記号が存在しない

□

$G$ をグラフ, $U \subset G$ を少なくとも1つ頂点を持つ部分グラフ

$U$ が順位ハンドル(precedence handle)である  $\stackrel{\text{def}}{\Leftrightarrow}$  全ての頂点に対し, $A, B$ でマーク付けられた頂点  $a, b$ とラベル $m$ の辺  $(a, b)$ が以下の条件を満たす

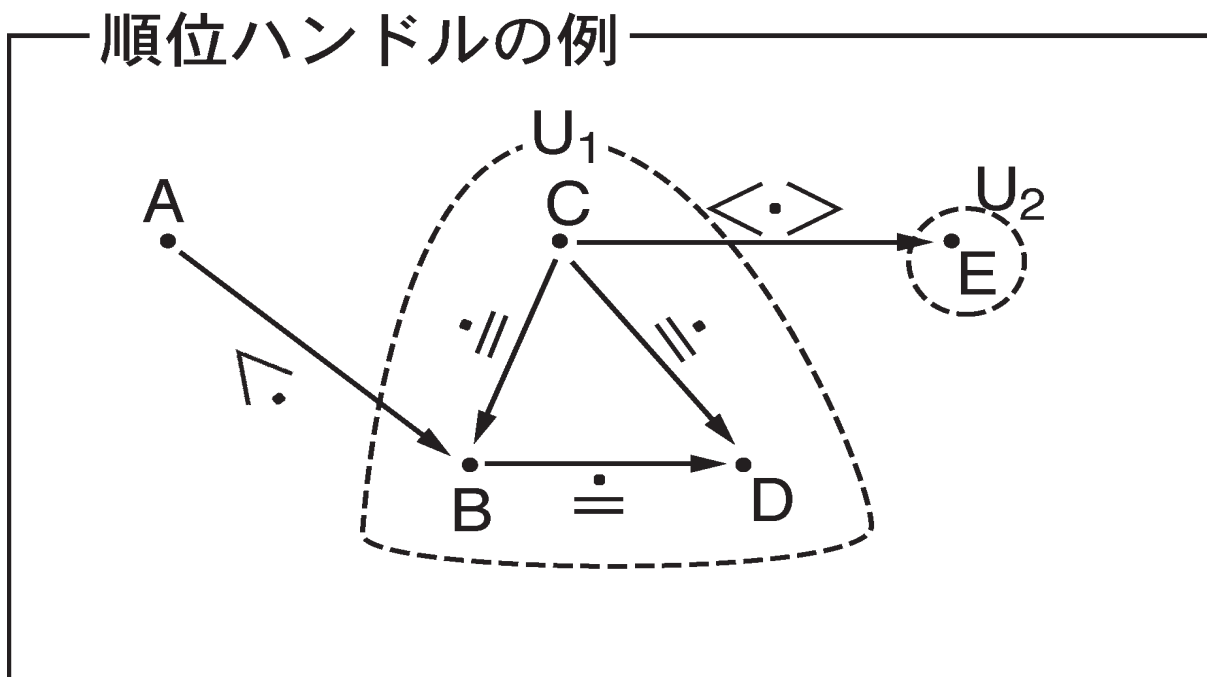
(i)  $a, b$ が $U$ に属するならば  $(A, B) \in \dot{=} m$

(ii)  $a$ が $U$ に属し, $b$ が属さないならば

$$(A, B) \in \cdot > m \cup < \cdot > m$$

(iii)  $b$ が $U$ に属し, $a$ が属さないならば

$$(A, B) \in < \cdot m \cup < \cdot > m$$



□

# Hiform96の様式 A1

プロジェクトコード:	A
プログラム名:	プログラム概要書
ライブラリ登録コード:	版名:
著作者:	初版発行日:
文書責任者:	現行版発行日:
キーワード:	CR分類コード:
目的・範囲:	
背景情報:	
記述言語:	所要ソフトウェア:
操作:	所要ハードウェア:
関連文書:	
機能:	
例:	

1. 文脈自由生成規則の作成
2. 順位関係表の作成
3. 構文解析が可能か検証

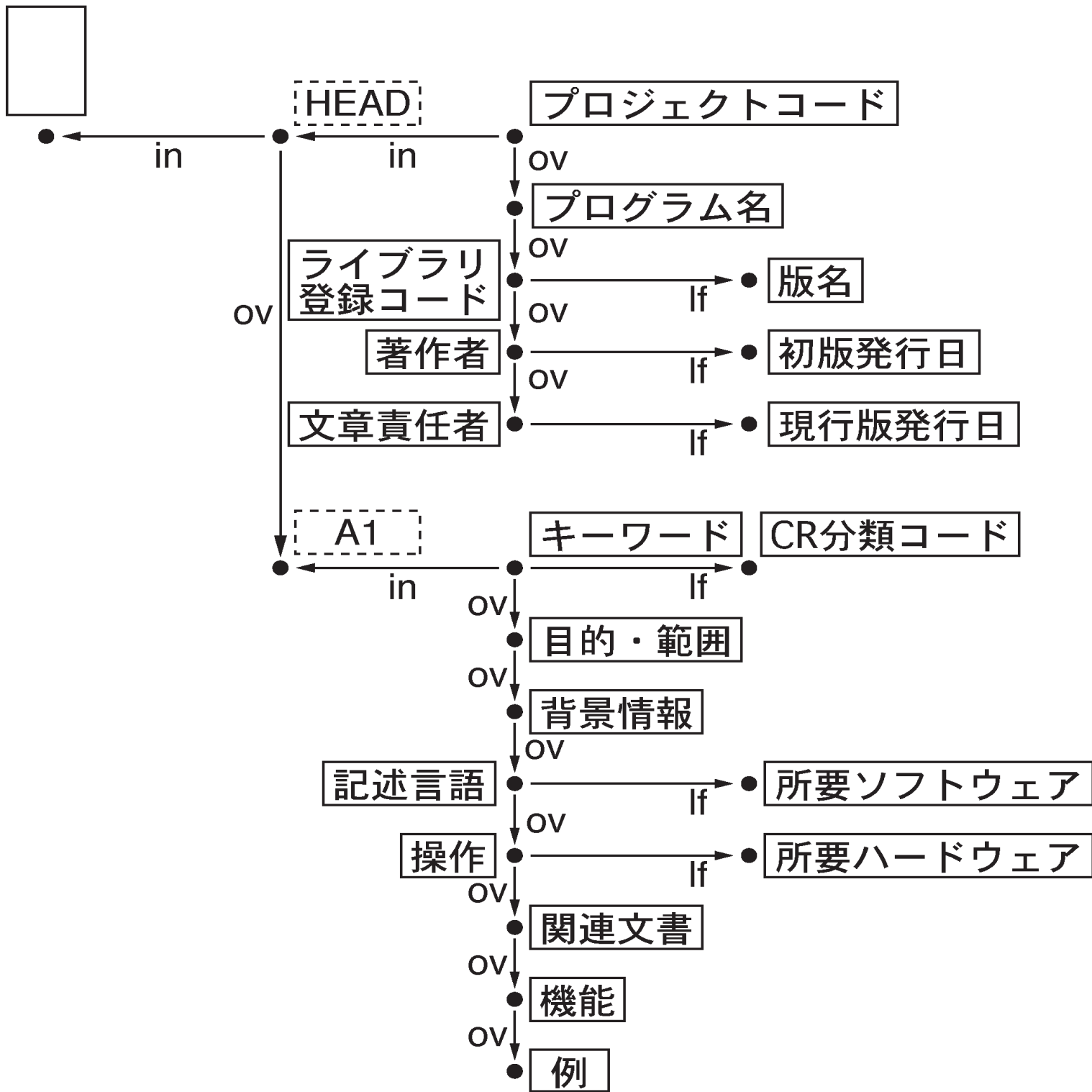
## 3.1 文脈自由生成規則の作成

様式 A1 を導出する規則は以下を考慮し作成

- 規則数を少なくした
- Label 数を最低限の数にした (3 個:in,over,left of)
- 様式共通部分 (head) と様式独自部分 (body) を分割

以上から作成した規則を次に示す

# 様式 A1 を表すマーク付きグラフ

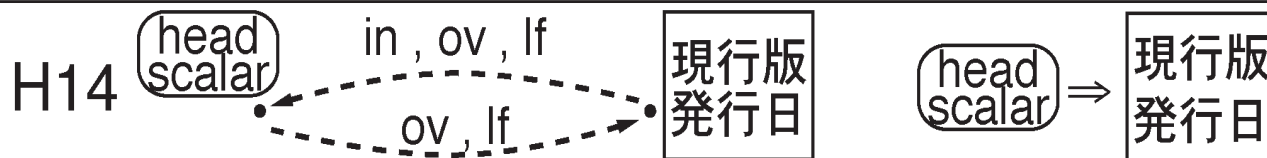
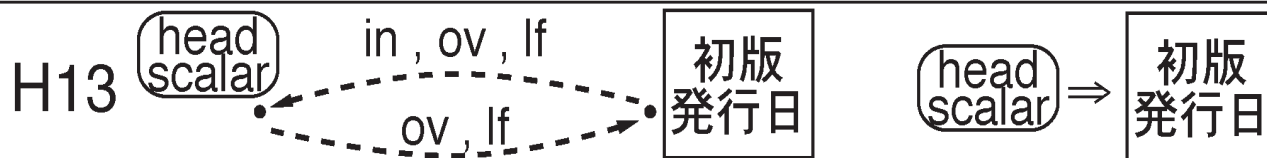
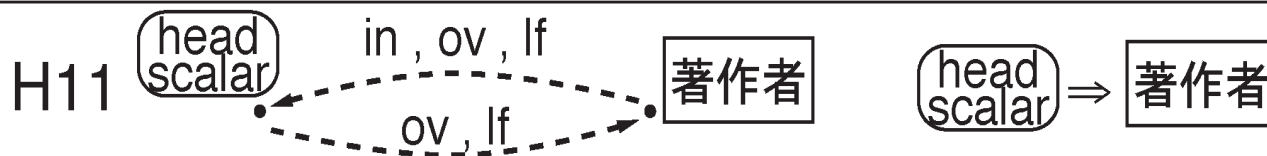
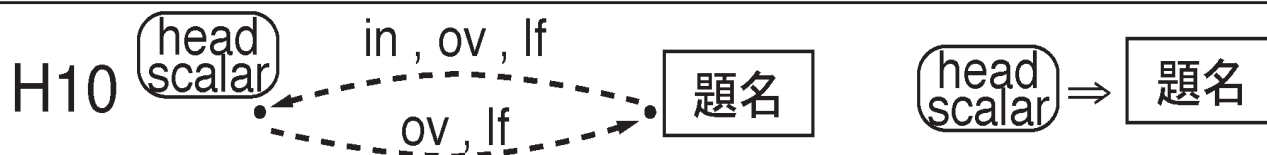


# 生成規則 (共通部分) 図 1

No. of the rule	Description of the rule	Picture for the rule
1		
2		
H1		
H2		
H3		
H4		
H5		
H6		

## 生成規則 (共通部分) 図2

No. of the rule      Description of the rule      Picture for the rule





# 生成規則 (A1 独自部分) 図 1

No. of the rule	Description of the rule	Picture for the rule
A1-0		
A1-1		
A1-2		
A1-3		
A1-4		
A1-5		
A1-6		
A1-7		
A1-8		

# 生成規則 (A1 独自部分) 図 2

No. of the rule	Description of the rule	Picture for the rule
A1-9		
A1-10		
A1-11		
A1-12		
A1-13		
A1-14		
A1-15		
A1-16		
A1-17		



### 3.2 順位関係表の作成

以下のように順位関係表を作成した

1. 作成した規則に基づき, 定義を用いて  
順位関係を定める
2. 定義 16 を元に正しいか否か検証

#### 生成規則数と順位関係表のサイズ

	A1 共通部分	A1 独自部分	合計
生成 規則数	$2 + 14 = 16$	18	34
順位関係表 サイズ	$14 \times 42 + 7 \times 21 = 735$	$17 \times 51 = 867$	1602

順位關係表 1

Left \ Right	共通部分 終端記号		head scalar		head column		head row		head root		HEAD		head		
	in	ov	If	if	in	ov	If	if	in	ov	If	if	in	ov	If
共通部分 終端記号		$\langle \cdot \rangle$	$\langle \cdot \rangle$			$\langle \cdot \rangle$	$\langle \cdot \rangle$			$\langle \cdot \rangle$	$\langle \cdot \rangle$			$\langle \cdot \rangle$	
head scalar		$\langle \cdot \rangle$	$\langle \cdot \rangle$			$\langle \cdot \rangle$	$\langle \cdot \rangle$			$\langle \cdot \rangle$	$\langle \cdot \rangle$			$\langle \cdot \rangle$	
head column		$\langle \cdot \rangle$	$\langle \cdot \rangle$			$\langle \cdot \rangle$	$\langle \cdot \rangle$			$\langle \cdot \rangle$	$\langle \cdot \rangle$			$\langle \cdot \rangle$	
head row		$\langle \cdot \rangle$				$\langle \cdot \rangle$				$\langle \cdot \rangle$	$\langle \cdot \rangle$			$\langle \cdot \rangle$	
head root										$\langle \cdot \rangle$	$\langle \cdot \rangle$			$\langle \cdot \rangle$	
HEAD															
head															

## 順位関係表 2

	Right		A1 終端記号		a1 scalar		a1 column		a1 row		a1 root		A1		a1	
	in	ov	in	ov	in	ov	in	ov	in	ov	in	ov	in	ov	in	ov
Left	in	ov	in	ov	in	ov	in	ov	in	ov	in	ov	in	ov	in	ov
A1 終端記号		$\langle \cdot \rangle$		$\langle \cdot \rangle$				$\langle \cdot \rangle$		$\langle \cdot \rangle$				$\langle \cdot \rangle$		
a1 scalar		$\langle \cdot \rangle$		$\langle \cdot \rangle$		$\langle \cdot \rangle$		$\langle \cdot \rangle$		$\langle \cdot \rangle$				$\langle \cdot \rangle$		
a1 column		$\langle \cdot \rangle$		$\langle \cdot \rangle$		$\langle \cdot \rangle$		$\langle \cdot \rangle$		$\langle \cdot \rangle$		$\langle \cdot \rangle$		$\langle \cdot \rangle$		
a1 row		$\langle \cdot \rangle$				$\langle \cdot \rangle$		$\langle \cdot \rangle$		$\langle \cdot \rangle$		$\langle \cdot \rangle$		$\langle \cdot \rangle$		
a1 root												$\langle \cdot \rangle$		$\langle \cdot \rangle$		
A1																
a1																

順位関係表 3

Left \ Right	A1		HEAD		a1		body		head		inner-struct		[ ]	
	in	ov	in	ov	in	ov	in	ov	in	ov	in	ov	in	ov
A1														
HEAD	<◇				<◇								>◇	
a1														
body														
head	<◇				<◇								>◇	
inner-struct													≡◇	
[ ]													≡◇	

### 3.3 構文解析

構文解析は以下の手順で行われる

1. Hiform96 の様式を表すグラフを入力



2. 順位関係表をもとに還元すべきハンドルを探索



3. 生成規則表と比較しハンドルを還元



4. これらを繰り返し、構文解析を完了する

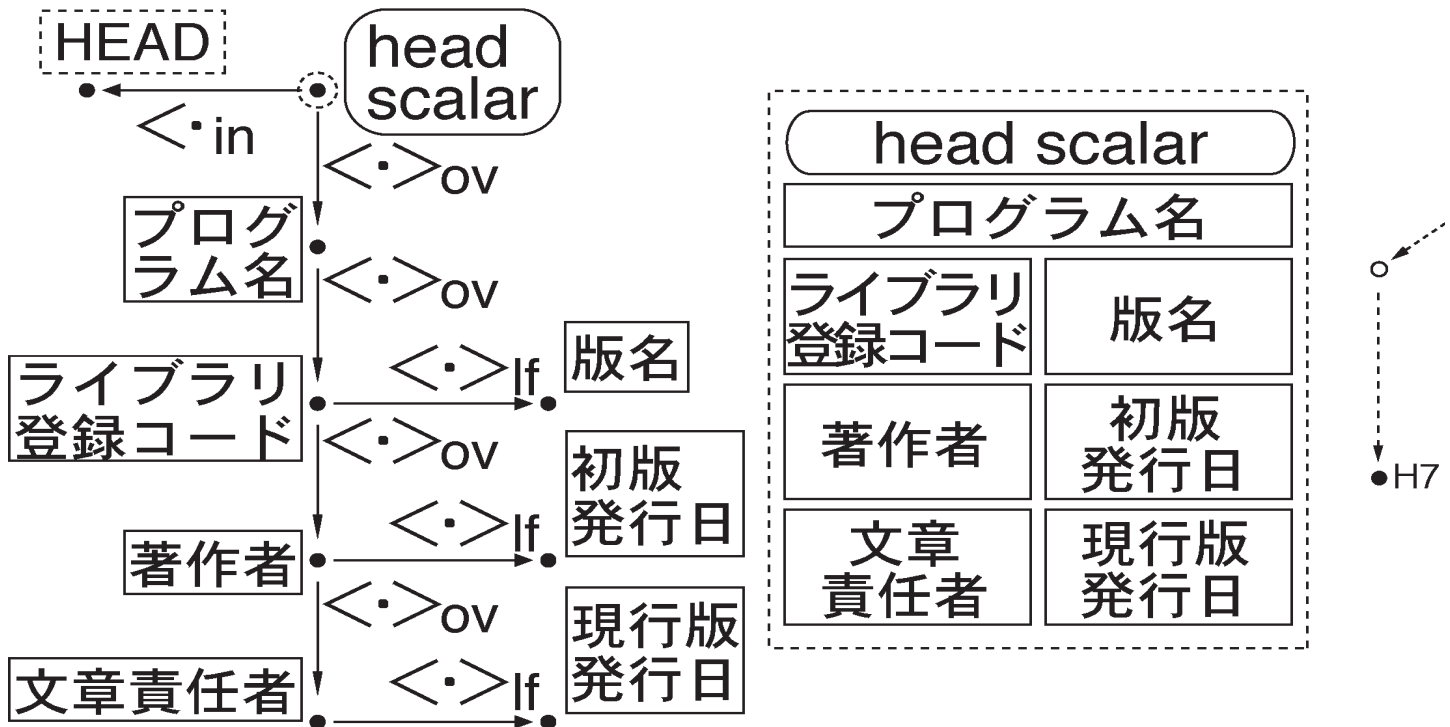
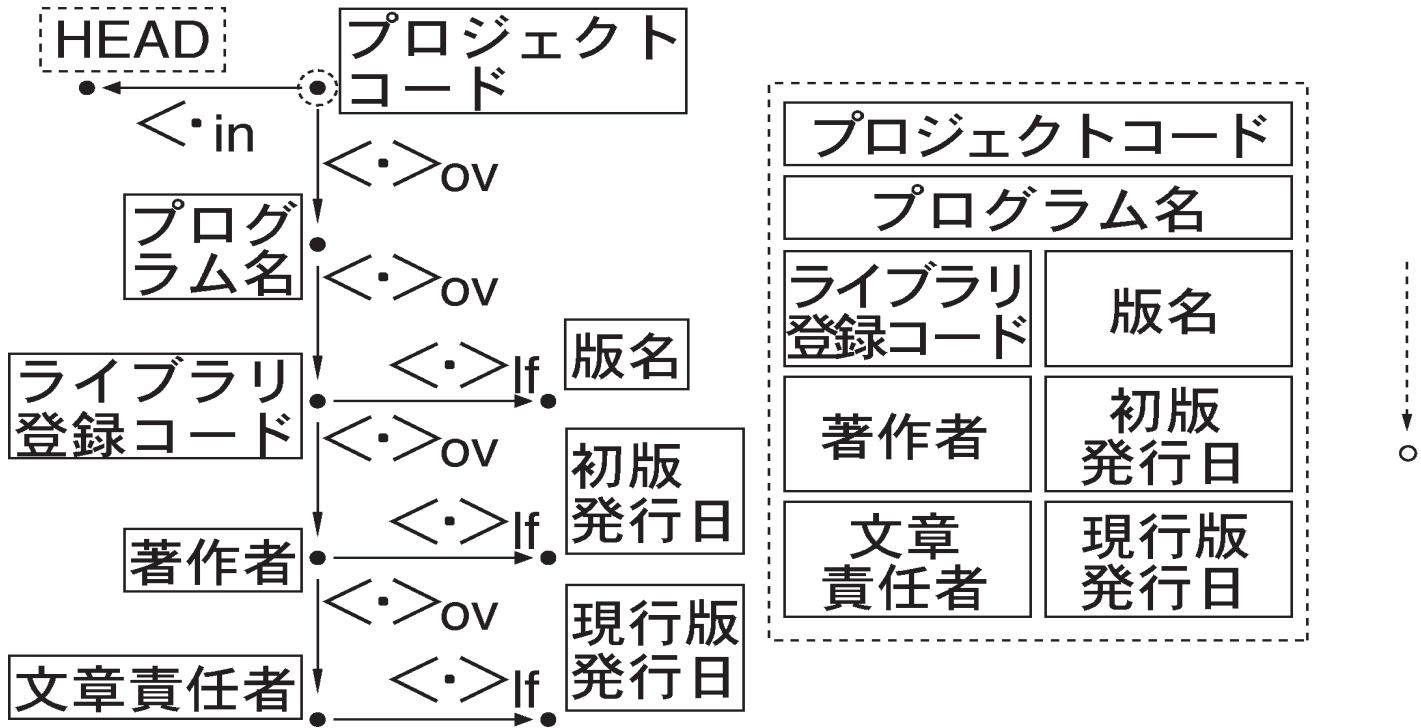


# 構文解析過程図(共通部分)1

マーク付きグラフ  
と順位関係

マーク付きグラフ  
の表示

直前の  
構文木

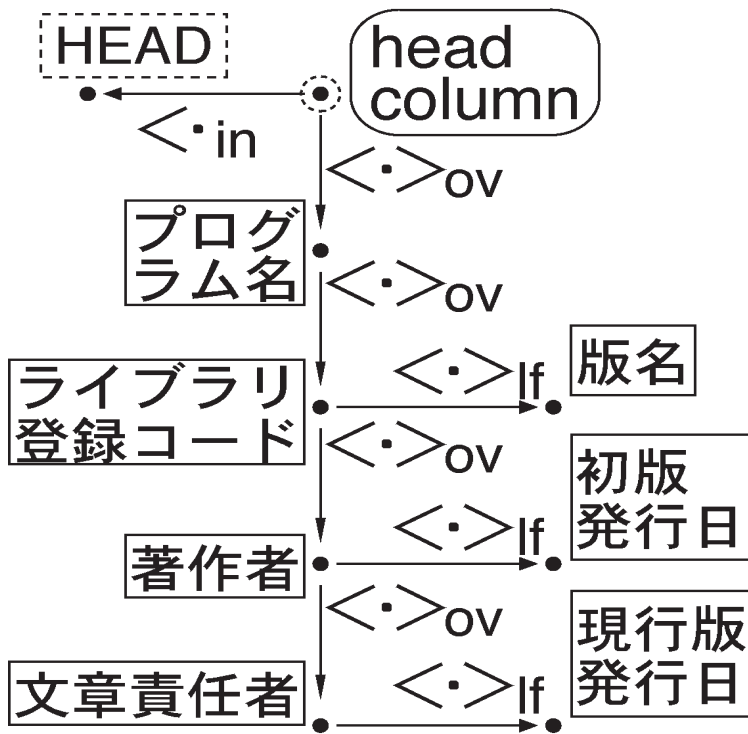


# 構文解析過程図(共通部分)2

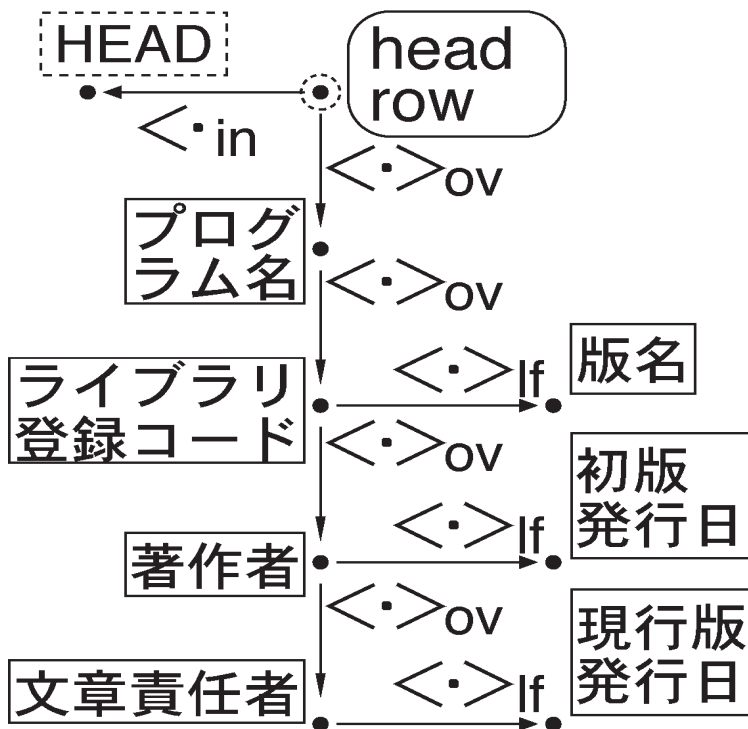
マーク付きグラフ  
と順位関係

マーク付きグラフ  
の表示

直前の  
構文木



head column	
プログラム名	
ライブラリ登録コード	版名
著作者	初版発行日
文章責任者	現行版発行日



head row	
プログラム名	
ライブラリ登録コード	版名
著作者	初版発行日
文章責任者	現行版発行日

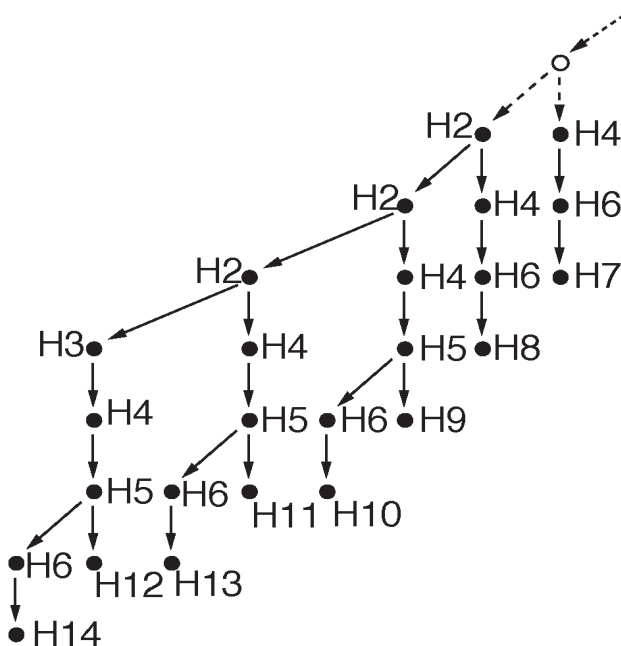
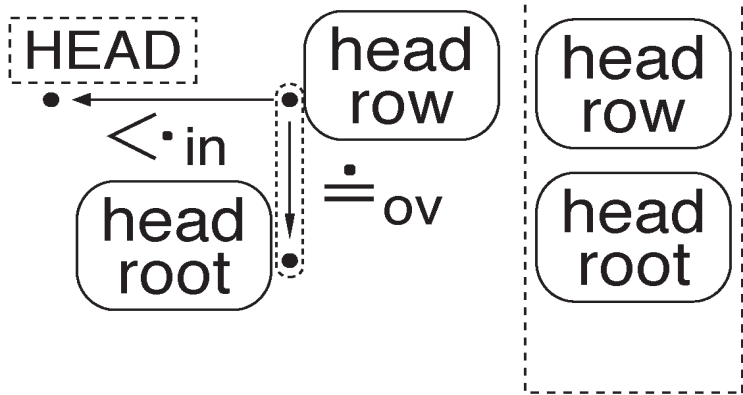
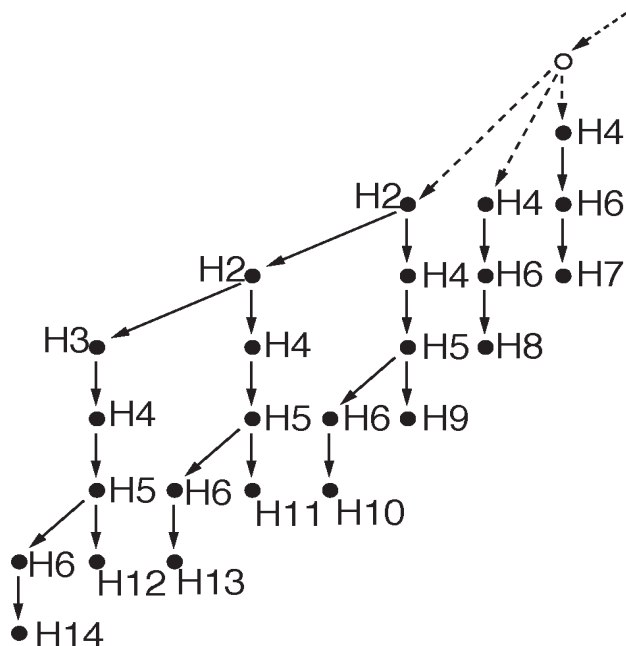
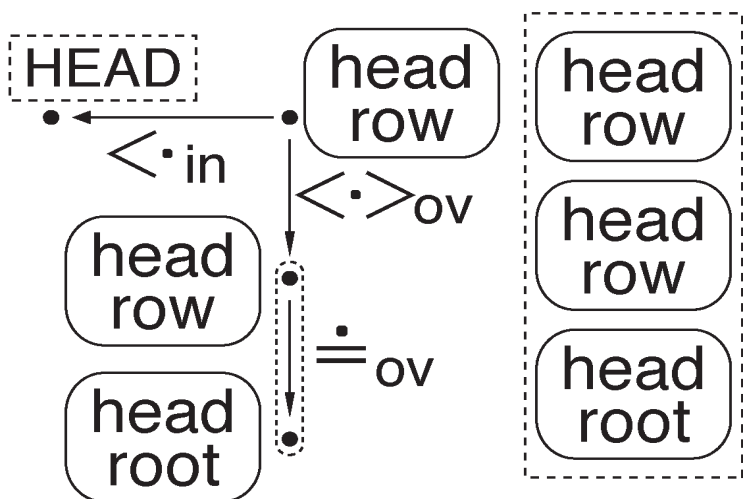


# 構文解析過程図(共通部分)3

マーク付きグラフ  
と順位関係

マーク付きグラフ  
の表示

直前の  
構文木

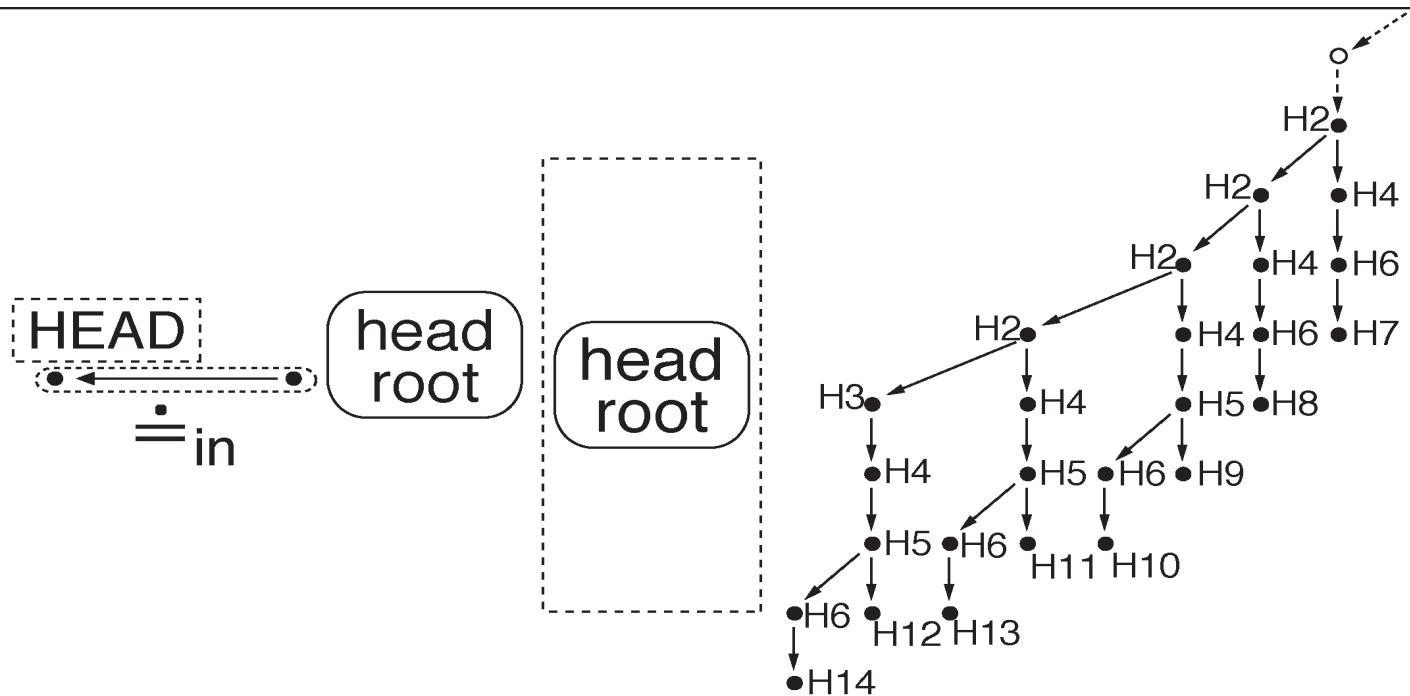


# 構文解析過程図(共通部分)4

マーク付きグラフ  
と順位関係

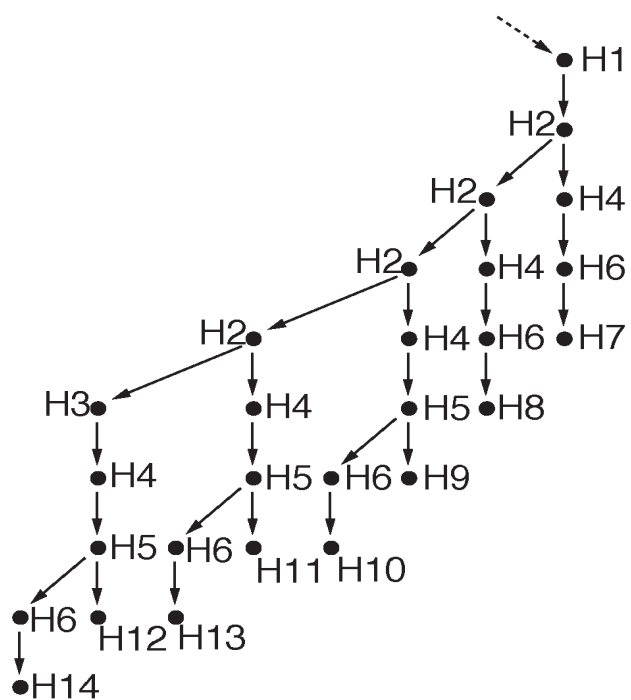
マーク付きグラフ  
の表示

直前の  
構文木



head  
•

head



## 主張

Hifrom96の様式A1に関して

1. 生成規則で生成可能である
2. 逆に生成規則と順位関係表により構文解析可能である

## 4. 今後の課題

1. 項目間の順序に関する自由度を下げる
2. 2次元の表のような格子状のグラフを扱う問題
3. 項目の大きさと位置を属性文法を用いて定める
4. 描画，文法に基づいての編集，計算機への実装