

# グラフ文法による図と表の処理 の定式化

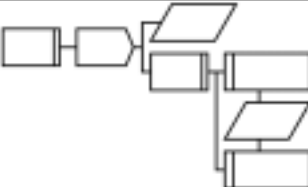
---

有田友和(日本大学)  
土田賢省(東洋大学)  
杉田公生(東海大学)  
夜久竹夫(日本大学)

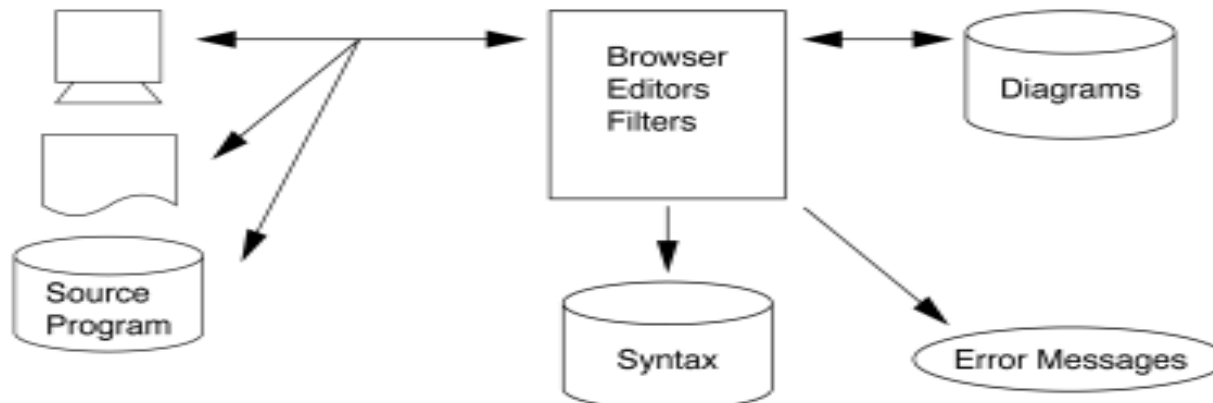
# 本発表: IFIP ICSE2000での発表とその後の我々の研究のまとめ

## 概要

### ■ TARGET: ソフトウェア仕様書に使用する図表

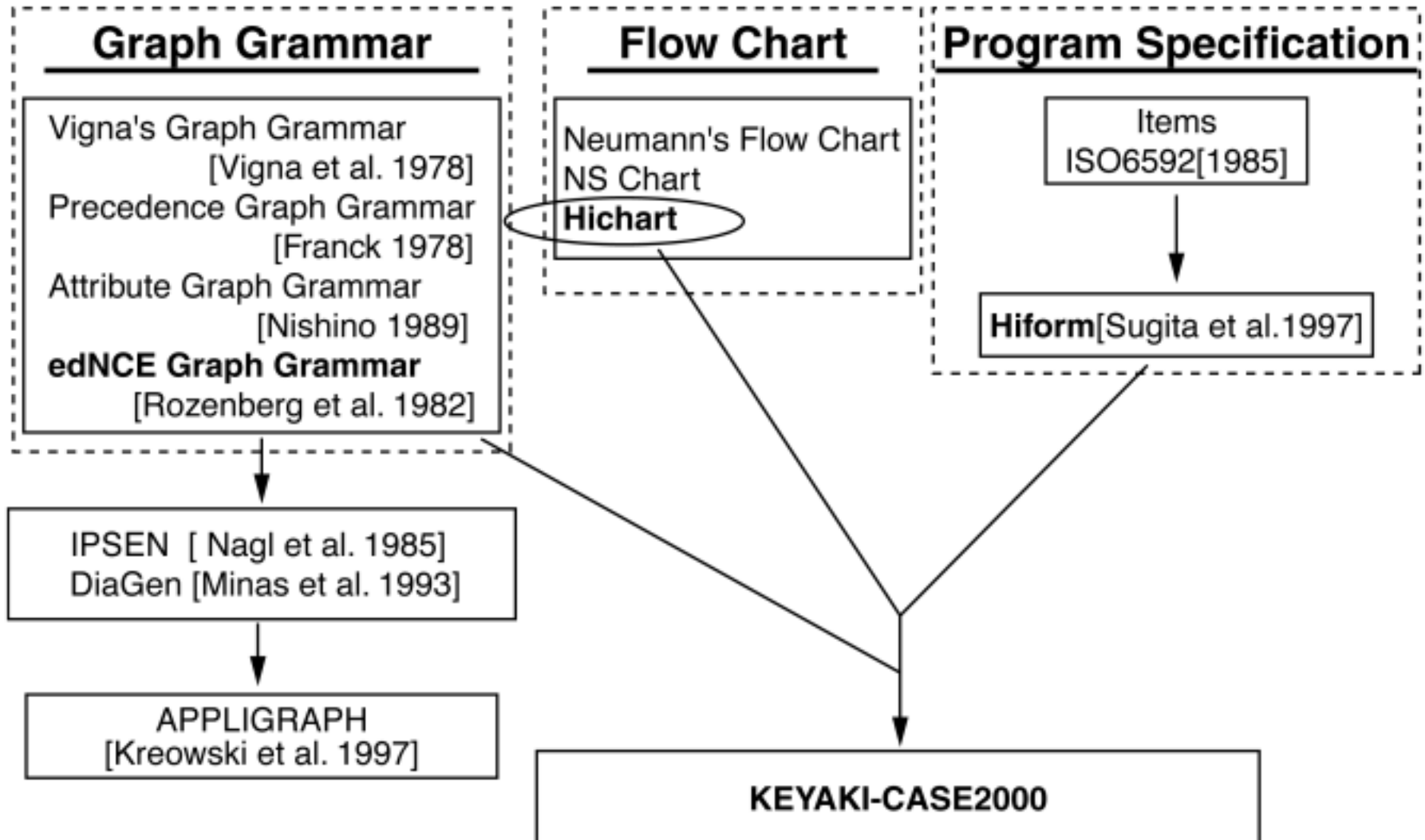
Diagrams	Corresponding Graphs	Universal Models												
Hierarchical Diagram 	Attribute Tree	Attribute NCE CFGG												
Nested Diagram <table border="1" data-bbox="694 606 1151 706"><tr><td>Program Code:</td><td>Program Specification</td></tr><tr><td>Program Name:</td><td>Version:</td></tr><tr><td>Library Code:</td><td></td></tr></table>	Program Code:	Program Specification	Program Name:	Version:	Library Code:		Attribute Marked Tree	Attribute NCE CFGG						
Program Code:	Program Specification													
Program Name:	Version:													
Library Code:														
Tessellation Diagram <table border="1" data-bbox="694 749 1151 849"><thead><tr><th>Name</th><th>Type</th><th>Size</th><th>G/L</th></tr></thead><tbody><tr><td>x</td><td>int</td><td>2</td><td>G</td></tr><tr><td>y</td><td>float</td><td>4</td><td>L</td></tr></tbody></table>	Name	Type	Size	G/L	x	int	2	G	y	float	4	L	Attribute Marked Tessellation Graph	Attribute NCE CSGG
Name	Type	Size	G/L											
x	int	2	G											
y	float	4	L											

### ■ GOAL: 図表の構文的処理システムの構築



# 1 はじめに

## 背景



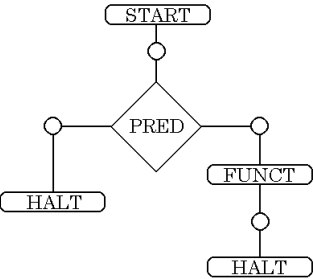
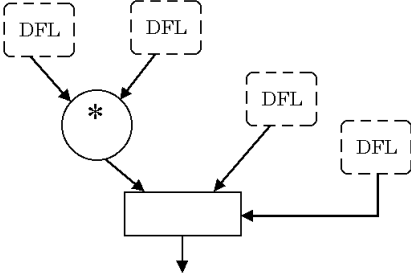
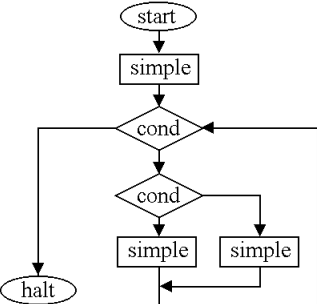


# Related Works

---

Project Name	Program Semantics by GG	Drawing by Combinatorial Algorithm	Drawing by AGG	Syntax by Graph Grammar
IPSEN	○	—	—	○
DiaGen	—	○	○	○
KEYAKI-CASE2000	—	○	○	○

## Related Works (continued)

Diagrams	Known Models	
<p data-bbox="91 287 373 339">Flowchart</p> 	<p data-bbox="795 287 1068 419">CF PLEX Grammar</p>	<p data-bbox="1348 287 1747 348">K.S.Fu (1982)</p>
	<p data-bbox="795 461 1081 594">Relational Grammar</p>	<p data-bbox="1348 461 1721 608">K.Wittenburg Et al.(1991)</p>
<p data-bbox="91 646 255 882">Data Flow Chart</p> 	<p data-bbox="795 646 1073 779">Positional Grammar</p>	<p data-bbox="1348 646 1753 793">G. Costagliola et al. (1990)</p>
<p data-bbox="91 989 388 1225">Structured Flow Chart</p> 	<p data-bbox="795 989 1264 1136">Symbol Relation Grammar</p>	<p data-bbox="1348 989 1690 1136">F.Ferruci et al. (1996)</p>



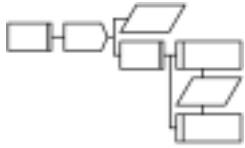
# 目的

---

- 図表の構造の構文的定義
- 図表のレイアウトに関する構文的定義
- 図表の構文を用いた処理方法(解析・編集等)
- 図表の構文的処理システム

# 結果

## ■ 文法による定式化 (図表の構造とレイアウト)

Graph Grammar	Grammar's type (Rewriting rule, Attribute rule)	Diagram												
HCGG	<i>Context-free, precedence</i> ( 67, 723 )													
HNGG	<i>Context-free, precedence</i> ( 280, 1252+280 )	<table border="1"><tr><td>Program Code:</td><td><b>Program</b></td></tr><tr><td>Program Name:</td><td><b>Specification</b></td></tr><tr><td>Library Code:</td><td>Version:</td></tr></table>	Program Code:	<b>Program</b>	Program Name:	<b>Specification</b>	Library Code:	Version:						
Program Code:	<b>Program</b>													
Program Name:	<b>Specification</b>													
Library Code:	Version:													
HTGG	<i>Context-sensitive</i> ( 69, 308 )	<table border="1"><thead><tr><th>Name</th><th>Type</th><th>Size</th><th>G/L</th></tr></thead><tbody><tr><td>x</td><td>int</td><td>2</td><td>G</td></tr><tr><td>y</td><td>float</td><td>4</td><td>L</td></tr></tbody></table>	Name	Type	Size	G/L	x	int	2	G	y	float	4	L
Name	Type	Size	G/L											
x	int	2	G											
y	float	4	L											

## ■ 図表の統一的処理方法

図表処理システム KEYAKI-CASE2000



# Contents

## 1. Introduction

## 2. Program Flowcharts and Program Specification Forms

Hichart

Hiform

## 3. An Attribute Graph Grammar for Hierarchical Diagrams

## 4. Attribute Graph Grammars for Tabular Diagrams

Nested  
Diagram

Tessellation  
Diagram

## 5. Diagram Processing System

HichartED   HiTS   LIVE

HiformED

## 6. Conclusion

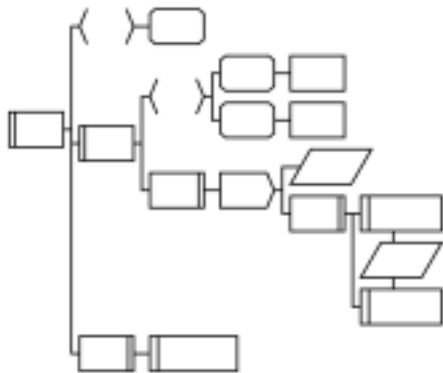


## 2. プログラム流れ図とソフトウェア仕様書

### Diagram in Program Specifications

#### Hichart

##### Hierarchical Diagram



#### Hiform

(Tabular Diagrams)

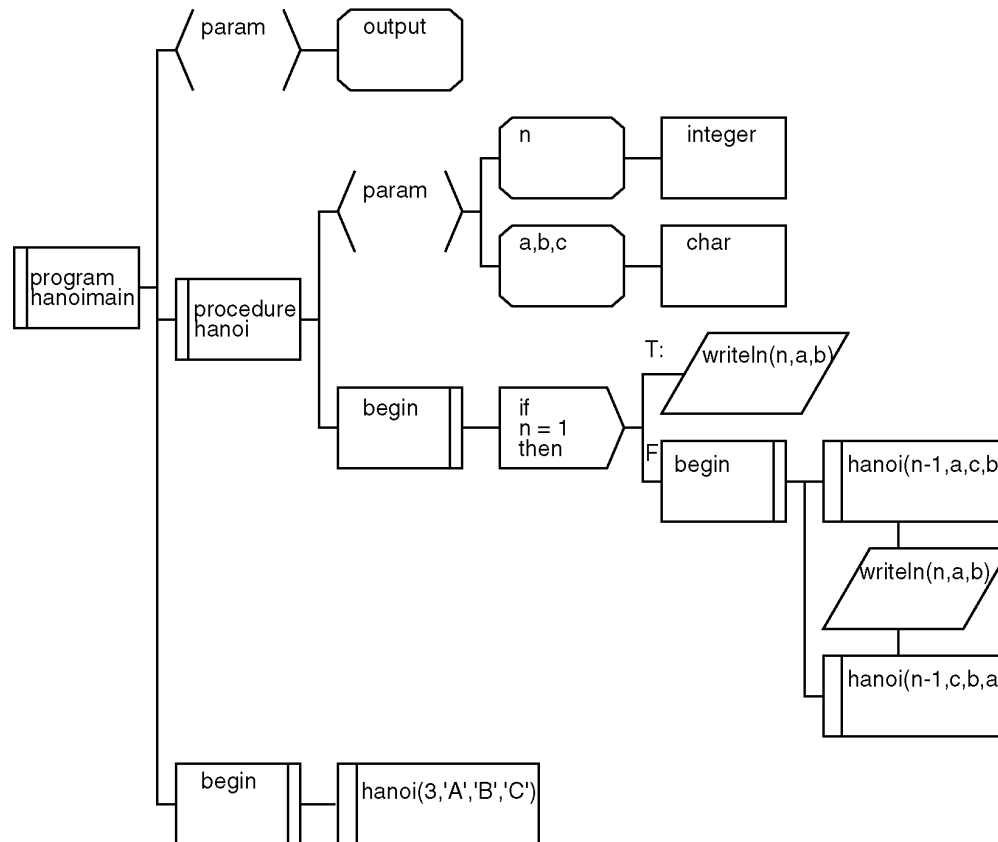
##### Nested Diagram

Program Code:	<b>Program Specification</b>
Program Name:	
Library Code:	Version:

##### Tessellation Diagram

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

## 2.1 階層型プログラム流れ図



Hichart プログラム流れ図 (ハノイの塔).

## 2.2表形式のプログラム仕様書

# Hiform

( a program specification language )

- ISO6592の項目を含む17種類の様式
- 表形式の仕様書

Project Code:	A 5
Program Name:	Program Specification-1 p
Library Code:	Version:
Author:	Original Release:
Approver:	Current Release:
Problem Description:	
Problem Supplementary Information (Theoretical Principles, Methods and References):	
Problem Solution: 1.Conventions and Terminology 2.Principles and Algorithms	

Project Code: hanoi_main	<b>A 5</b>
Program Name: procedure hanoi	<b>Program Specification-1</b> p
Library Code: cs-2000-01	Version: 2.1
Author: Tomokazu Arita	Original Release: 1999/12/25
Approver:	Current Release: 2000/1/31
Problem Description:	
How to describe the structure of this program.	
Problem Supplementary Information (Theoretical Principles, Methods and References):	
Theoretical Principles: Hichart Diagram	
Problem Solution:	
1.Conventions and Terminology 2.Principles and Algorithms	
1. Convention : Hichart Diagram	
<pre> graph LR     subgraph Program         P[program hanoi]     end     subgraph Procedure         H[procedure hanoi]         H --- Param1[param]         Param1 --- N{n}         Param1 --- ABC{a,b,c}         N --- Integer[integer]         ABC --- Char[char]         H --- Begin1[begin]         Begin1 --- If{if n = 1 then}         If --- TrueT[/writeln(n,a,b)/]         If --- FalseF[begin]         FalseF --- Hanoi1[hanoi(n-1, a,c,b)]         Hanoi1 --- TrueF[/writeln(n,a,b)/]         TrueF --- Hanoi2[hanoi(n-1, c,b,a)]     end     subgraph MainCall         BeginMain[begin]         BeginMain --- HanoiMain[hanoi(3, 'A', 'B', 'C')]     end     P --- H     P --- BeginMain   </pre>	

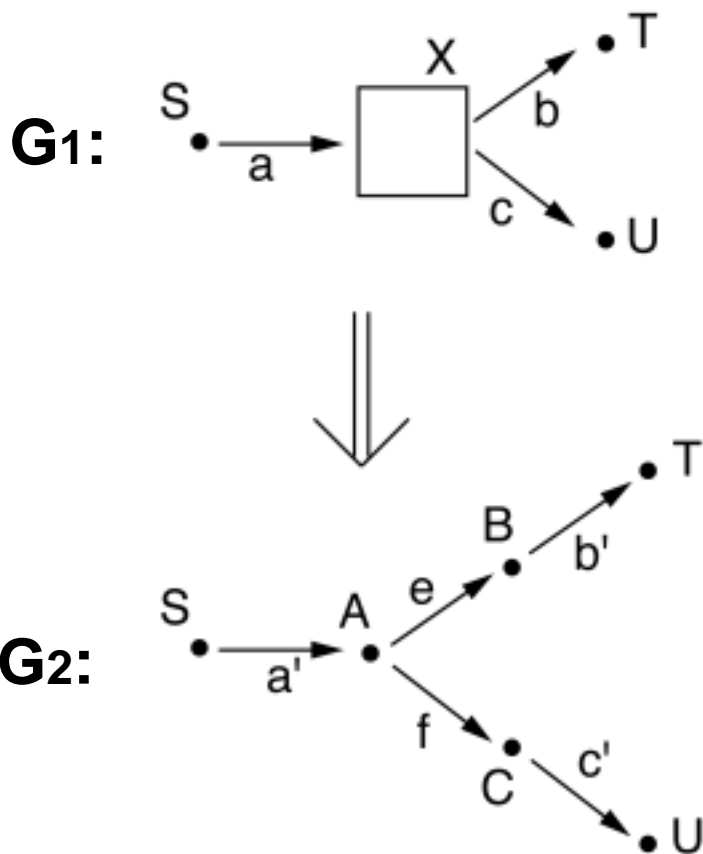


# 3. 階層型プログラム図を定式化する 属性グラフ文法

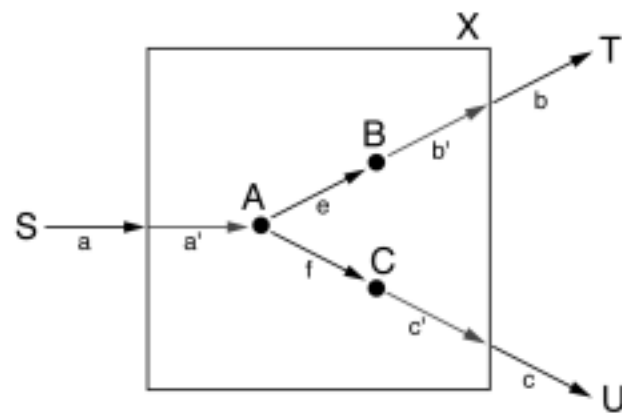
---

# NCE ( Neighborhood Controlled Embedding ) Graph Grammar [ Rozenberg et al. 1982]

導出例

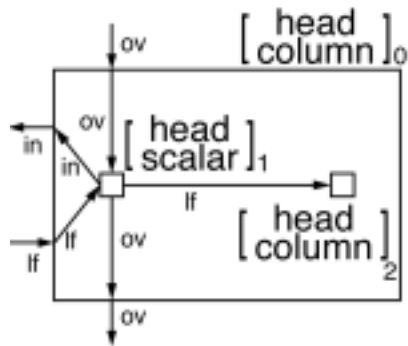


生成規則:



# Attribute Context-free NCE Graph Grammar

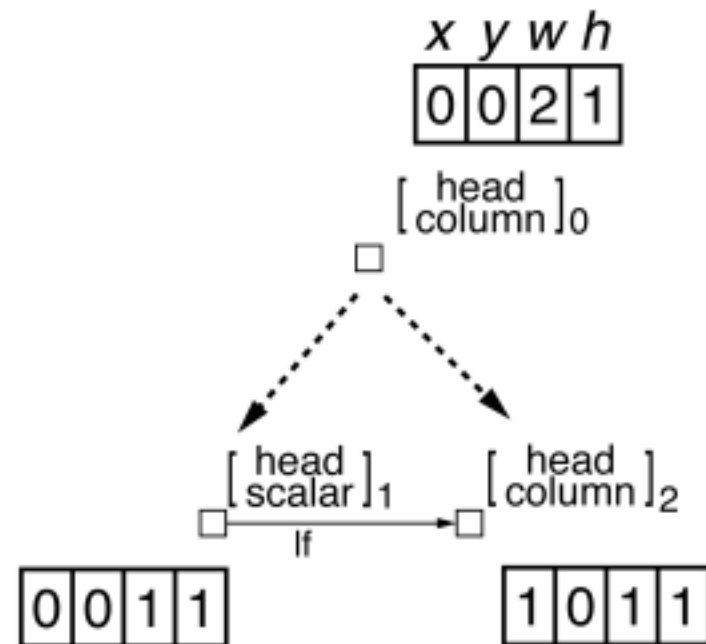
## 生成規則と属性規則



### Semantic Rule

$x(1) = x(0)$   
 $x(2) = x(0) + \text{width}(1)$   
 $y(1) = y(0)$   
 $y(2) = y(0)$   
 $\text{width}(0) = \text{width}(1) + \text{width}(2)$   
 $\text{height}(0) = \max(\text{height}(1), \text{height}(2))$

## 属性評価:



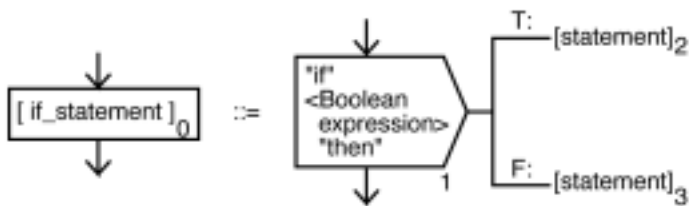
# ■ 文法 3.1 HCGG

**HCGG** (HiChart Graph Grammar):

Hichartプログラム流れ図に対する

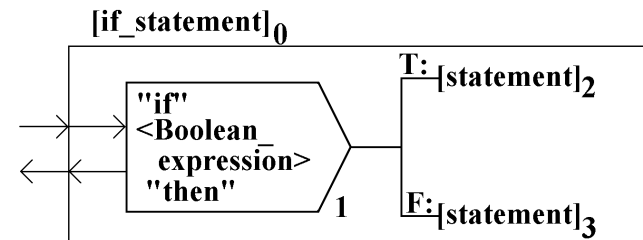
属性文脈自由NCEグラフ文法

Production Example of HCGG



if\_statement (1)

Production



Semantic Rules

$top(2)=top(0)$	$cl(2)=\"T:\"$
$top(3)=bottom(2)+GapY$	$cl(3)=\"F:\"$
$x(1)=x(0)$	$id(1)=id(0)$
$x(2)=x(0)+w(1)+GapX$	$id(2)=id(1)+1$
$x(3)=x(0)+w(1)+GapX$	$id(3)=id(2)+nc(2)$
$y(0)=(y(2)+y(3))/2$	$nc(0)=1+nc(2)+nc(3)$
$bottom(0)=max(bottom(1),bottom(3))$	

$w(1)=MinW$

$h(1)=get\_height([\"if\",<Boolean\_expression>,\"then\"])$

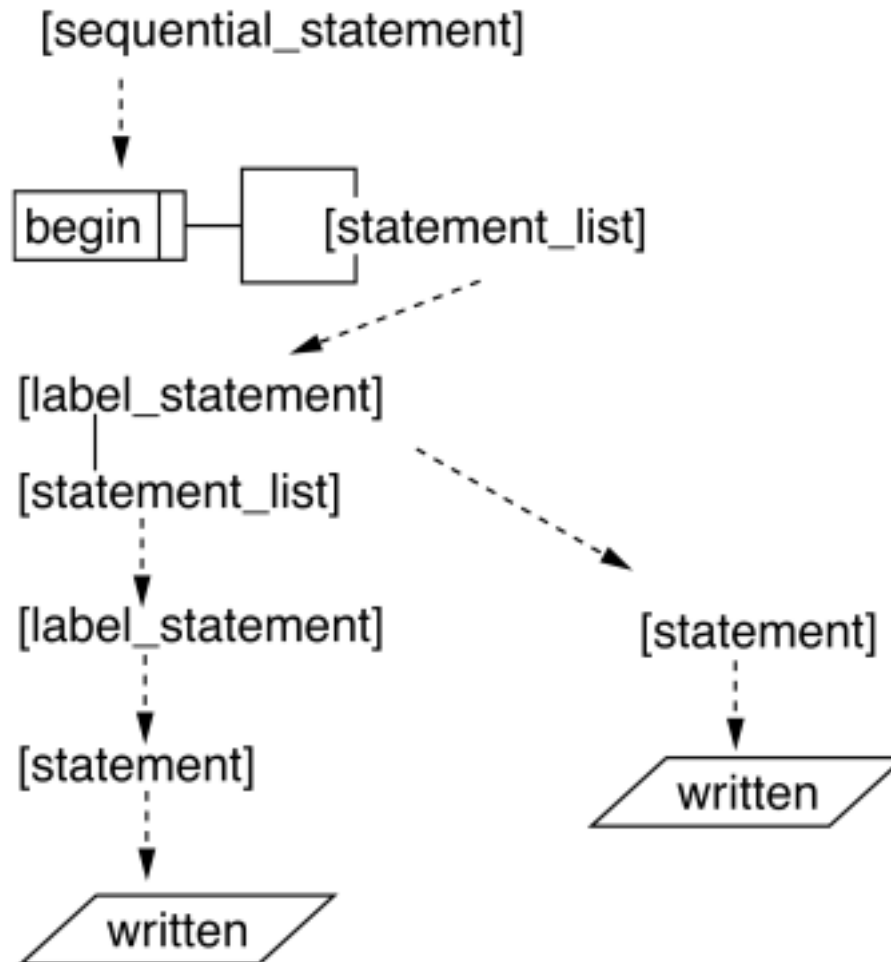
$cell(1)=\"exclusive\_selection\"$

$string(1)=get\_str([\"if\",<Boolean\_expression>,\"then\"])$

$lines(1)=get\_line(1,[2,3])$



# ■ HCGGの導出木



## HCGGの特徴

GG	Type	Rewriting Rule	Attribute Rule
HCGG	文脈自由	67	723



■ **命題 3.2**

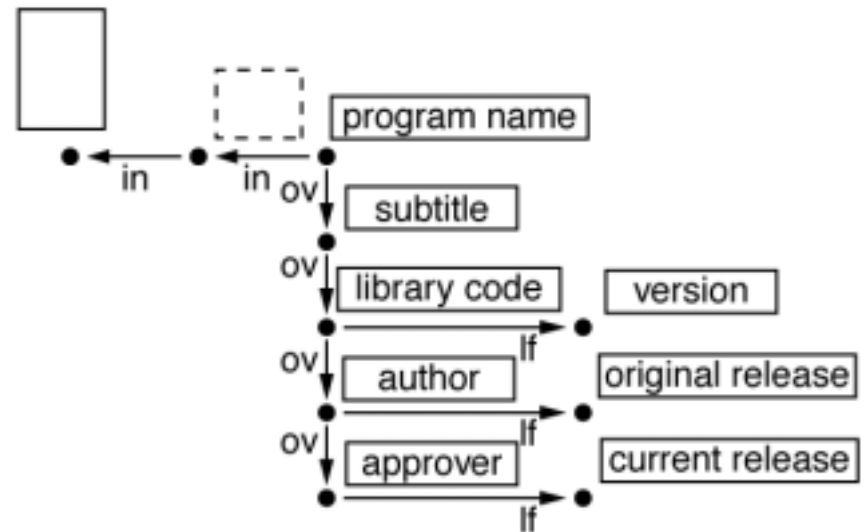
**HCGGは順位グラフ文法である.**

## 3.2 表に対する属性NCEグラフ文法

### ■ 表とそれに対応するマーク付きグラフ

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

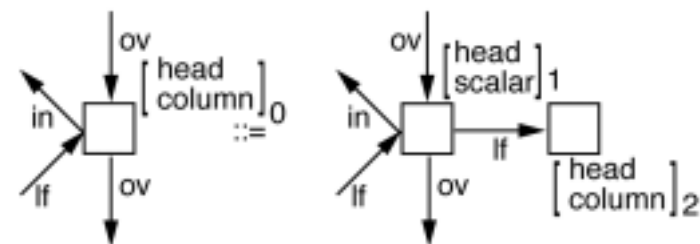
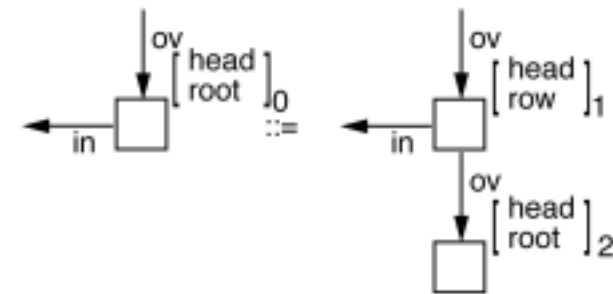
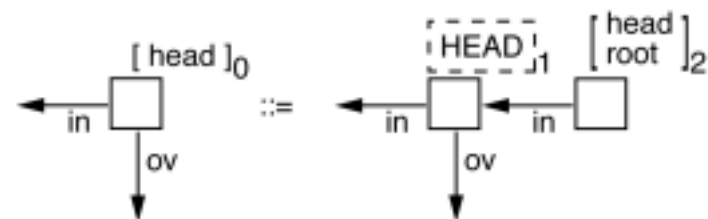
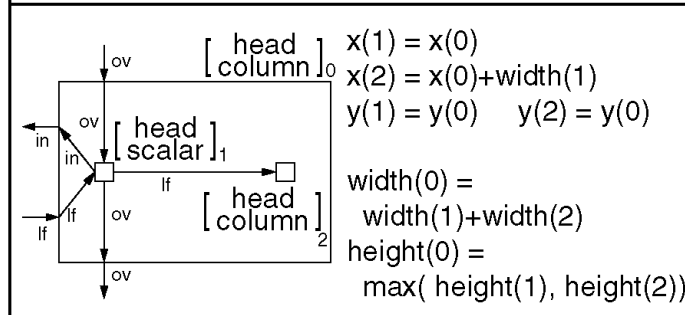
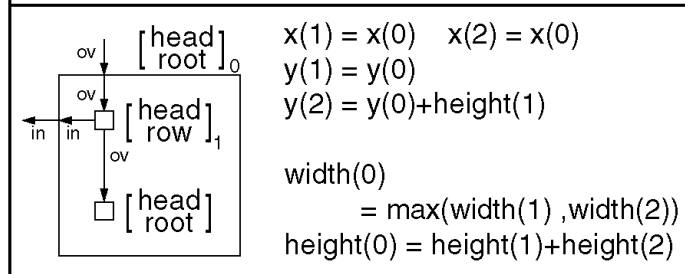
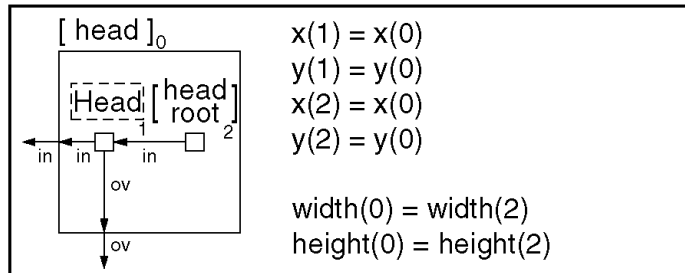
program name :	
subtitle :	
library code :	version :
author :	original release :
approver :	current release :



# ■ 文法 4.1 HNGG

## HNGG (Hiform Nested Graph Grammar) : 表形式の仕様書Hiformを定式化する 属性NCEグラフ文法

### Production Examples of HNGG



## HNGGの特徴

<b>GG</b>	<b>Type</b>	<b>Rewriting Rule</b>	<b>Attribute Rule</b>
HNGG	文脈自由	280	1252+280



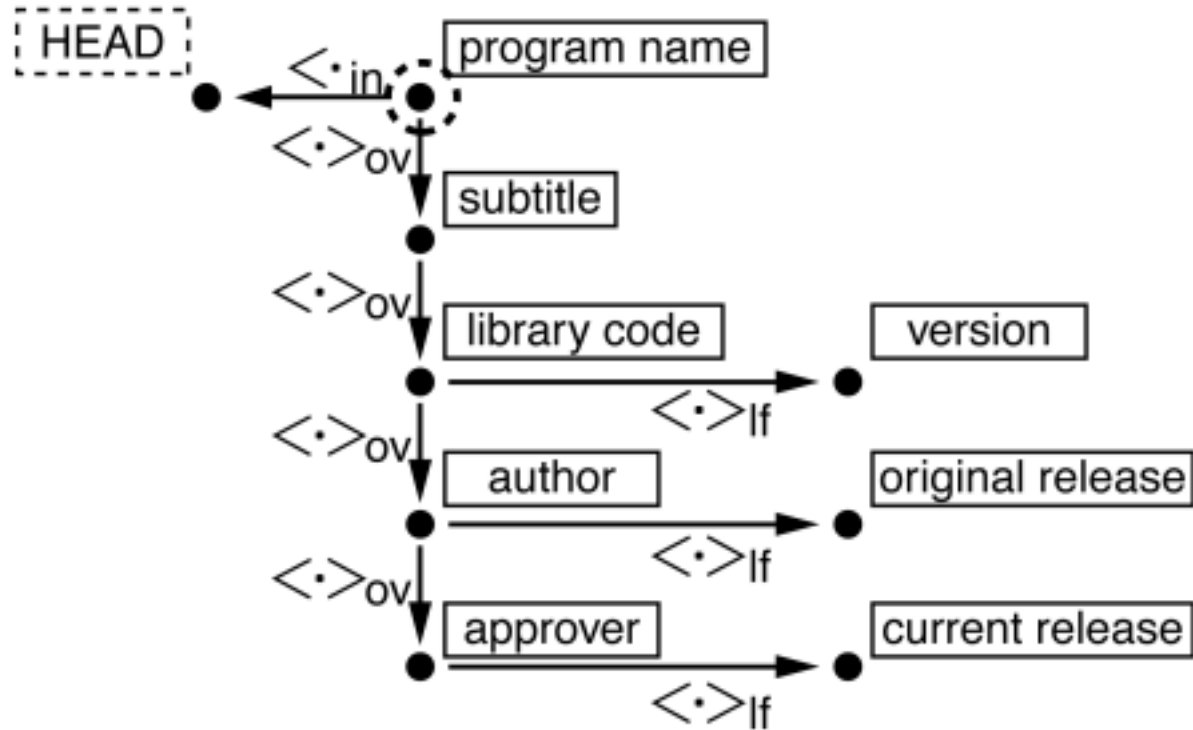


■ 命題 4.2

HNGG は順位グラフ文法である.

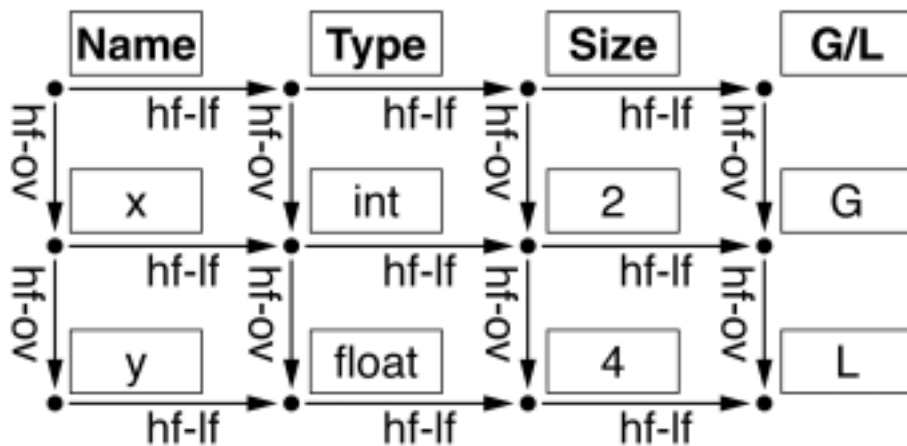


## ■ How to use precedence rule



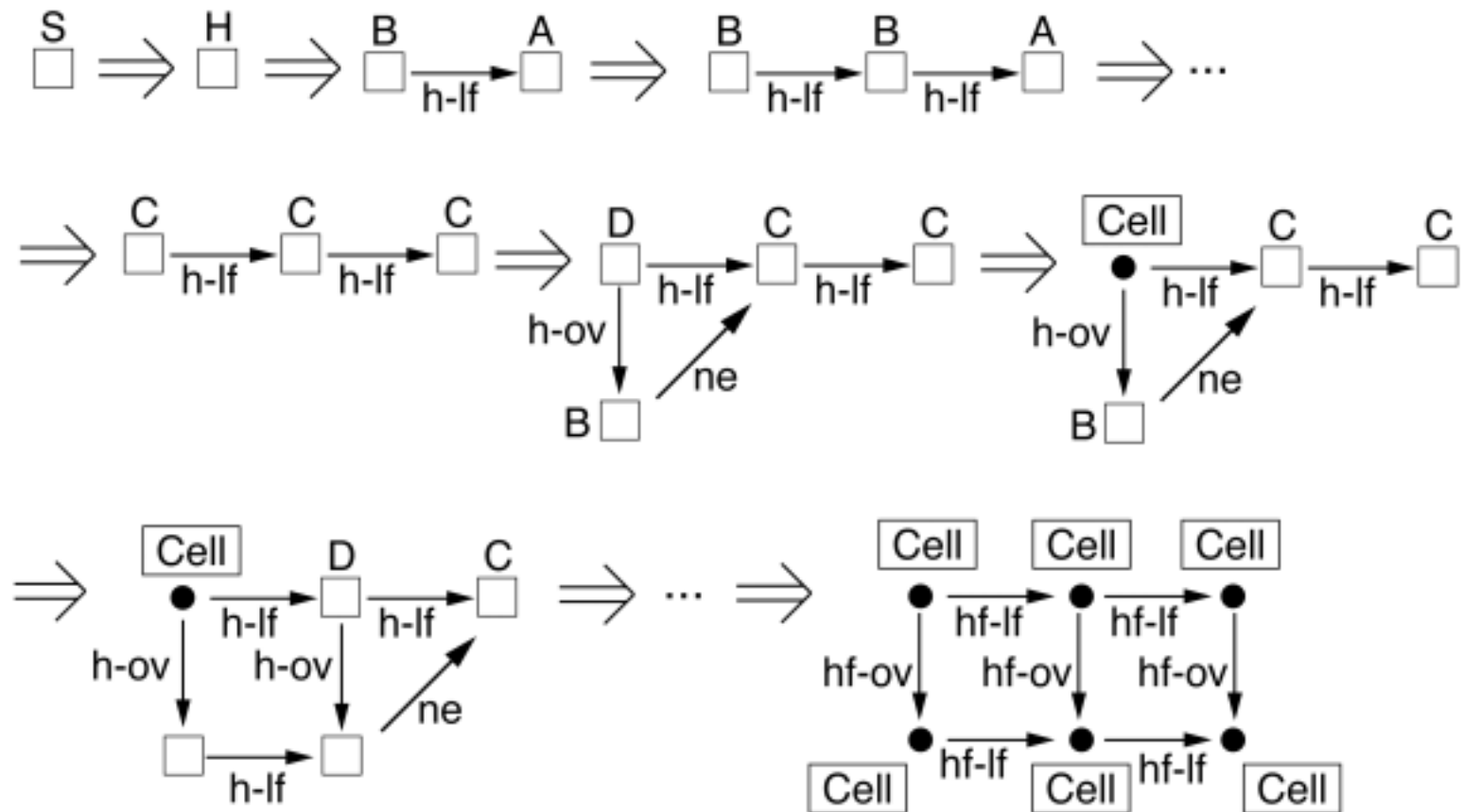
## ■ 格子状の表とそれに対応するグラフ

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





# ■ HTGGの導出



## HTGGの特徴

GG	Type	Rewriting Rule	Attribute Rule
HTGG	文脈依存	69	308





## 5. 図表処理システム

---

### **KEYAKI – CASE2000 Concept**

#### **1. HichartED**

Hichart プログラム図編集コンポーネント

#### **2. HiTS**

Hichart プログラム図フィルタリングコンポーネント

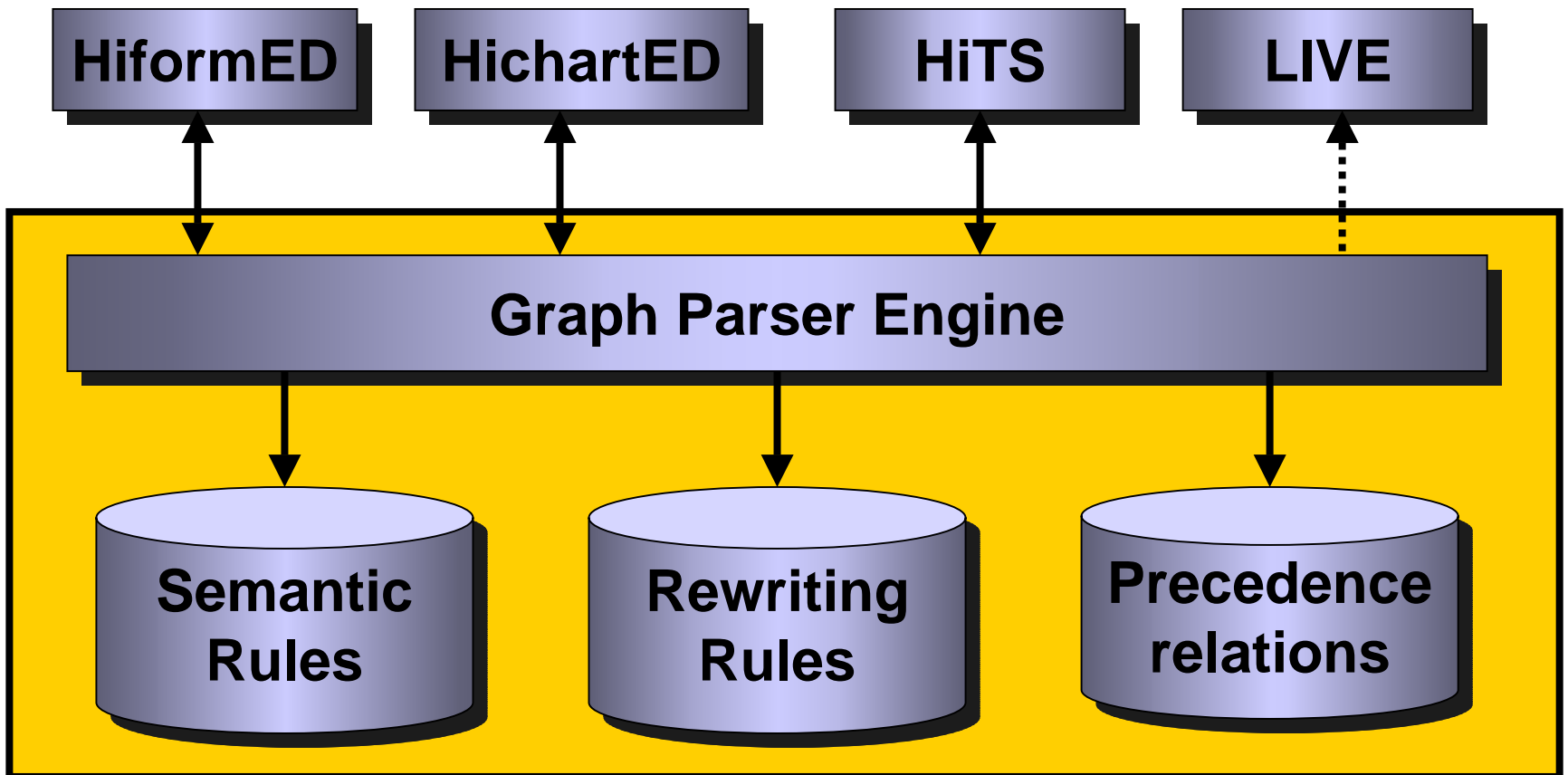
#### **3. LIVE**

プログラム変数解析コンポーネント

#### **4. HiformED**

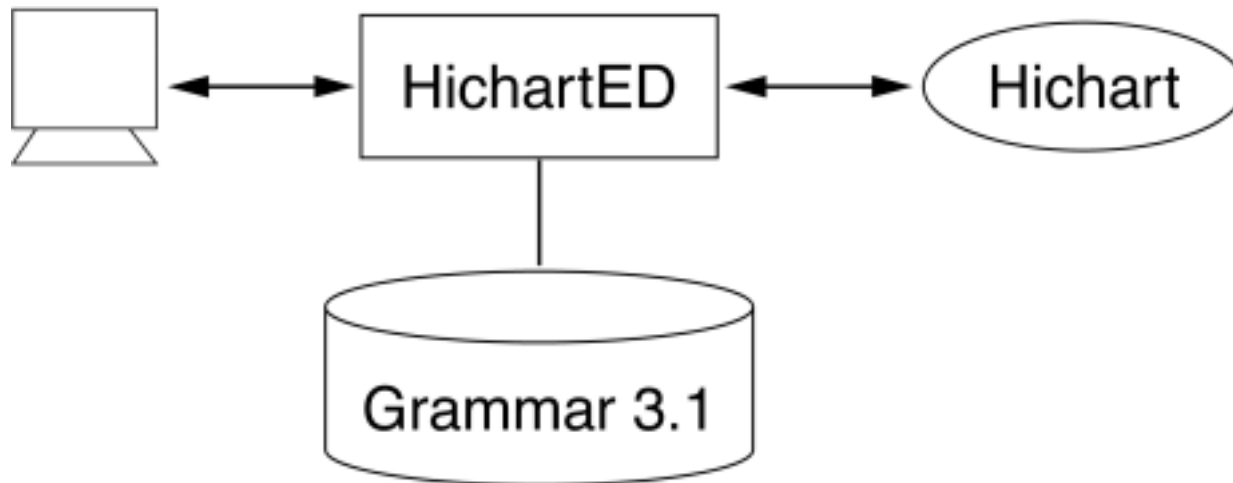
Hiform プログラム仕様書編集コンポーネント

# Keyaki-case 2000



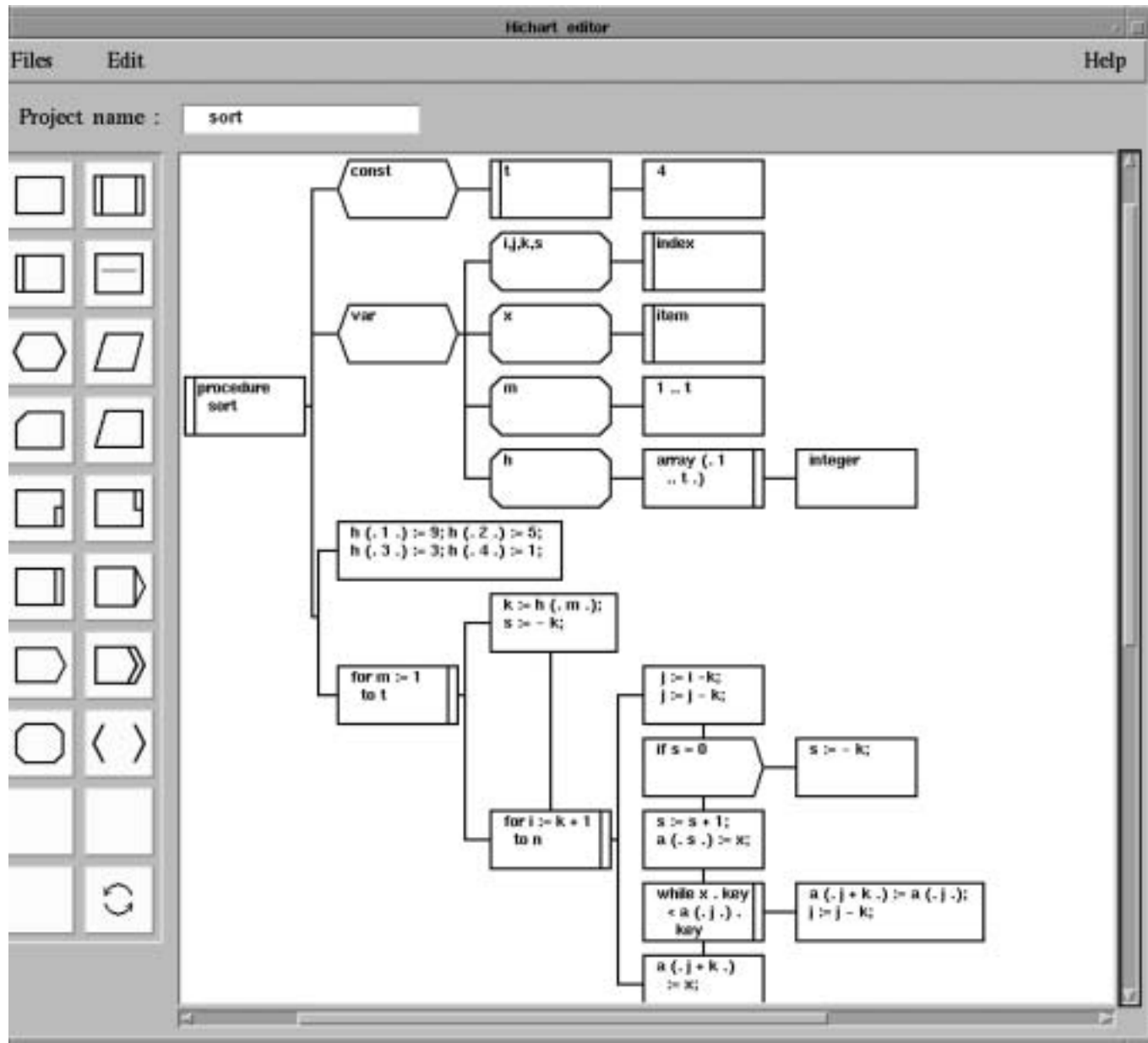
## 5.1 HichartED

- Hichart流れ図作成・編集コンポーネント
- 構文指向型エディタ
- 編集方法は構文に基づいて定式化





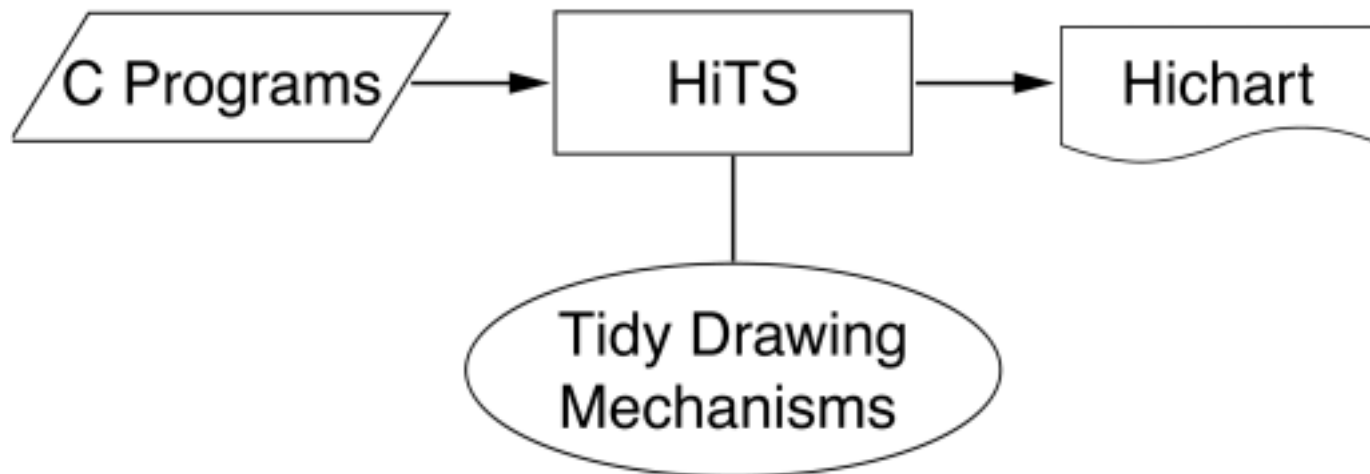
# Screen Concept of HichartED



## 5.2 HiTS

Miyadera et.al [7,9]

- プログラム流れ図フィルタリングコンポーネント
- ソースからプログラム図を生成する.
- 木構造図の美的描画処理による描画
- データ構造や処理の流れの視覚化支援



# Execution Screen in HiTS

The screenshot shows the HiChart Translation Service interface. On the left is a control panel with the following elements:

- Language selection: English / Japanese
- Input a target source file:
- Select Layout type:
- Input Cell's size: horizontal length  vertical length
- Select objects if you need:
  - Picture
  - HiChart flowchart (.pc file)
  - Abstract HiChart flowchart (.a file)
  - C-code
  - Abstract H-C-code
  - Asm code
- 

At the bottom of the control panel, it says: Computer Software Lab / Data Base Division, [masahiro@iis.fuji.wu.ac.jp](mailto:masahiro@iis.fuji.wu.ac.jp)

The main execution screen on the right is divided into two sections:

- Top section:** A large flowchart diagram with various nodes and arrows. Two large, hand-drawn eyes are superimposed on the top right of this section.
- Bottom section:** A code editor window showing the following C code:

```
#include <stdio.h>
#include <limits.h>

typedef enum {FALSE, TRUE}
BOOL;

#define DF INT_MAX
#define N 8
#define START 0

int lr[N], ld[N];

void di_extra(int start,
(
    BOOL alist[N];
    int i, current, s;

    for (i = 0; i < N; i++)
        alist[i] = I
        ld[i] = I
        lr[i] = -

    alist[start] = TR
    count = i
    ld[start] = 0;

    do {
        sin = DF
        for (i =
            ld[i] < sin) {
                ...
            }
        }
    }
}
Auto save file
```

## 5.3 LIVE

Miyadera et.al. [9]

- プログラム変数解析コンポーネント
- プログラムモジュール化を支援



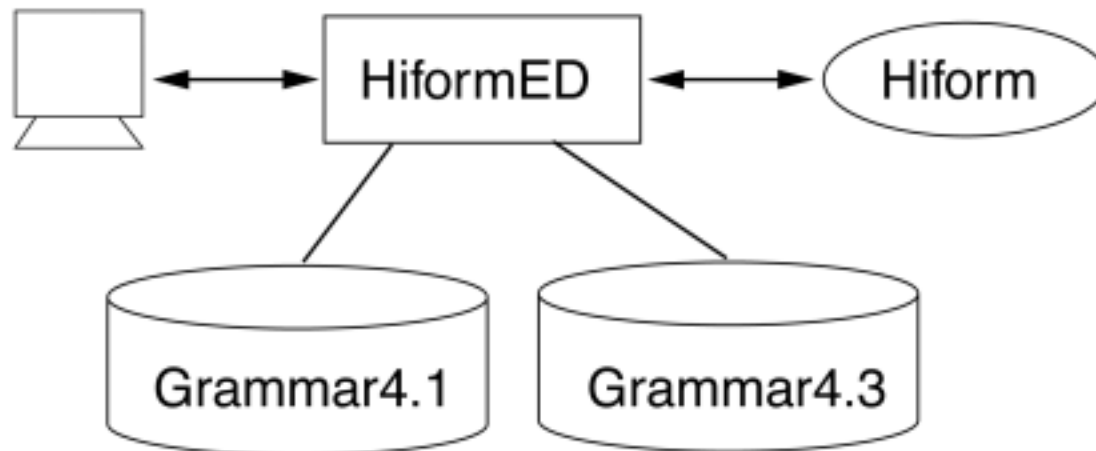
# Execution Screen in LIVE

The screenshot displays the LIVE (Live Implementation of Visual Execution) environment. It features several windows and components:

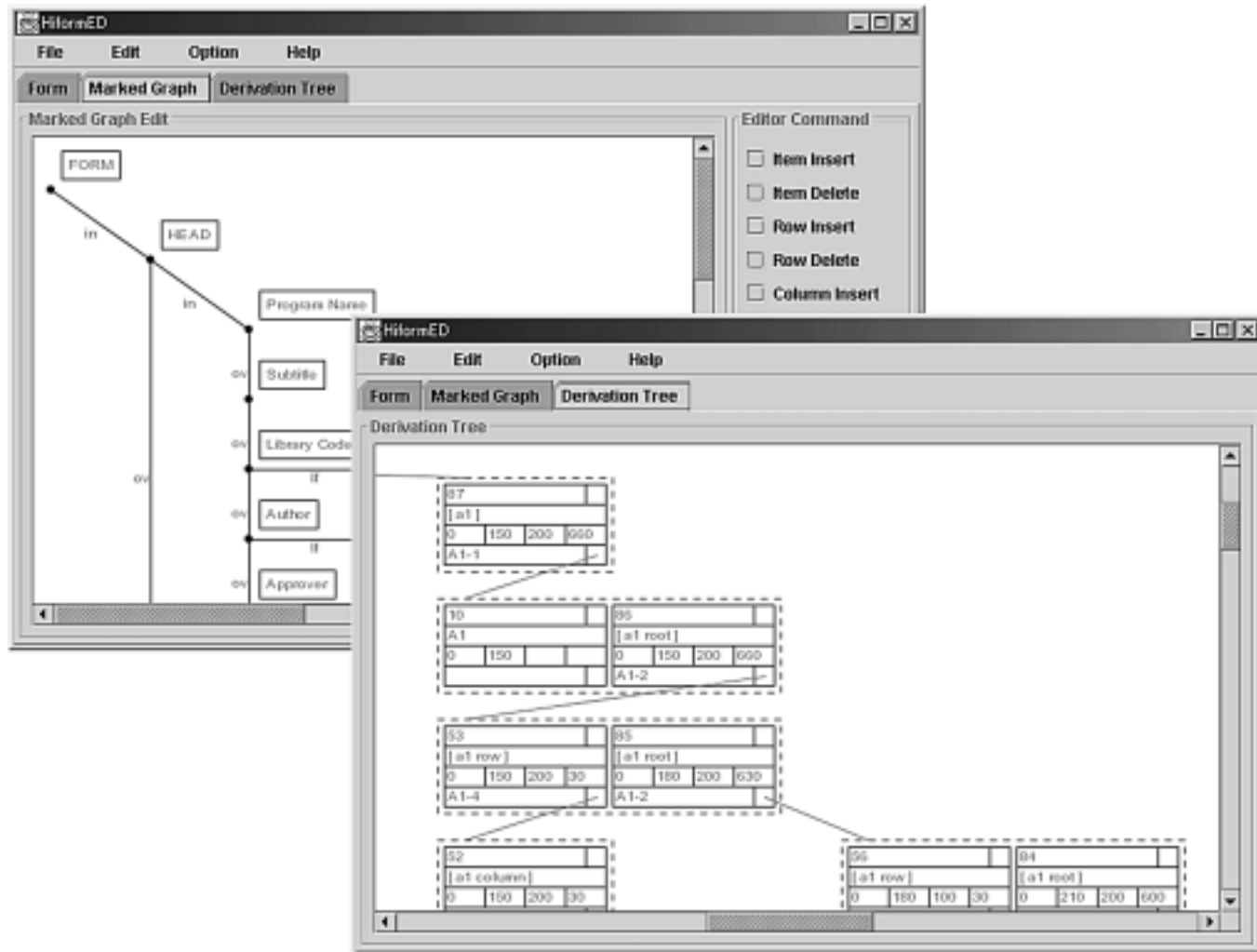
- Top Left:** A circular gauge and a waveform graph.
- Top Center:** A window titled "ファイル" (File) showing a file browser with a list of files including "matrisoul\_liv", "matrisoul prt", "matrisoul prt paper17", "rcvsw.p", "screen1-1.ps", and "screen1.ps". A search filter is applied, and buttons for "了解" (Info), "フィルター" (Filter), and "取消" (Cancel) are visible.
- Left Panel:** A window titled "File" containing the source code for a program named "Matrisoul". The code includes constants, variable declarations, and nested loops for reading and writing matrix elements.
- Bottom Center:** A window titled "File" showing the execution progress, with line numbers 00010 through 00048 and corresponding execution status indicators.
- Right Panel:** A large window titled "Data Flow Graph" (DFG) showing a complex network of nodes and edges, representing the data flow of the program's execution.

## 5.4 HiformED

- プログラム仕様書Hiform編集システム
- 文法4.1に基づくパーサーは開発済み
- 文法4.1に基づいた編集方法は一部定式化 [有田, GT-VMT02]
- 属性評価によるXMLソースの出力可能  
XML Viewerによる閲覧可能 [井上等, IASTED AI2002 ]



# Screen Concept of HiformED





# 結論

---

- 図表の構造を属性グラフ文法により定式化した。
  - 図表のレイアウトを属性グラフ文法により定式化した。
  - 図表の統一的処理方法を提案した。
- 
- 統一的処理システムは現在開発中



## 6. 結論(つづき)

System	NCE GG	System with NCE GG	System without NCE GG
HichartED	△	△	○
HiTS	○	Not yet	○
LIVE	Not yet	Not yet	○
HiformED	△	△	△

# Our Project Web Site :

Including detailed description of Graph Grammars

■ **URL:**

<http://www.hichart.org/>