File Structures for Modular Form Processing Systems

日本大学大学院　総合基礎科学研究科
地球情報数理科学専攻　博士前期課程

仲川　俊一

修士論文発表会　2002年2月19日

Contents

- **1章** Introduction
  対象, 背景, 歴史, 目的など
- **2章** Known Results
  表とグラフ, 構文解析, 内部コード
- **3章** System Structure
  システム構造, データ構造
- **4章** File Format
  FXL
- **5章** Conclusion

1. Introduction

1.1 Target

Tabular Form and Graph Grammar

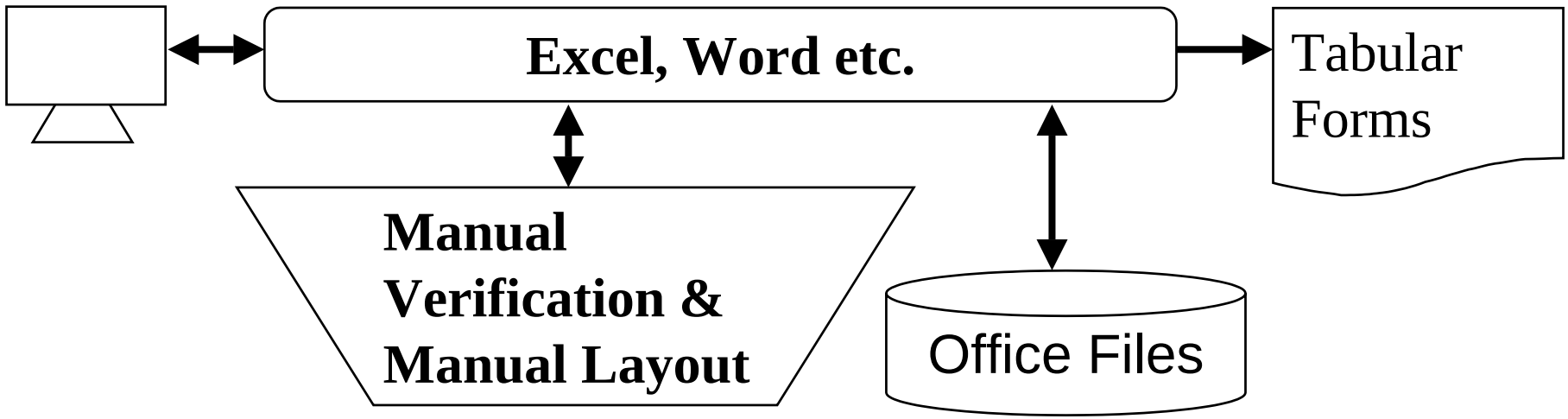
Program Name:	
Subtitle:	
Library Code:	Version:
Author:	Original Release:
Approver:	Current Release:
Problem Description:	
Problem Supplementary Information (Theoretical Principles, Methods and References):	
Problem Solution: 1.Conventions and Terminology 2.Principles and Algorithms	

Program Specification Form

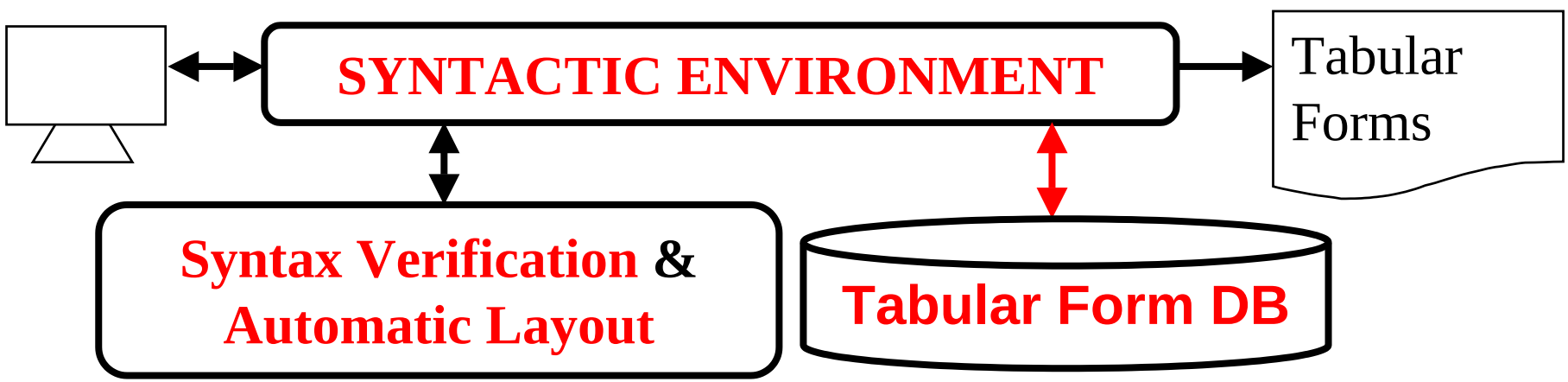
- **To Guide Syntactically Valid Items by Productions**
- **To Evaluate the Impact Area of Rewriting by Productions**
- **Automatic Drawing by Attribute Rules**

1.2 Background and Position

Present : Manual Designing



Our Goal : Syntactic Designing



1.3 Background 2

Program Diagrams

Hichart, PAD, SPD, HCP (1980's)

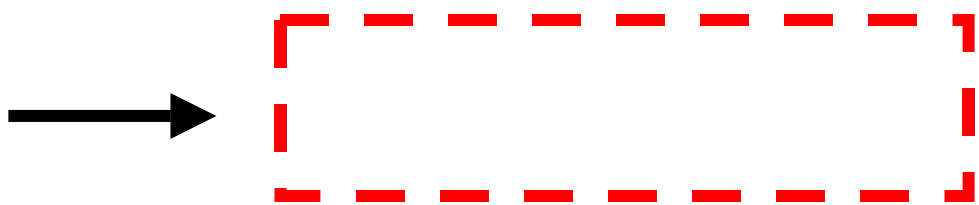
DXL code (BNF) (1995)

H-Code2 (BNF) (1997)

Program Specification

ISO6592 (1985)

Hiform (1997)



1.4 Motivation

- Common file formats among tabular form processing systems are required. (cf. DXL)

1.5 Purpose

- To survey present status of modules and protocols (Section 2)
- To determine a system structure for all processing system (Section 3)
- To determine file formats of the system (Section 4)

1.6 Results

1. We provide short summary, illustrations and examples related. (Section 2)

2. We determine the system structure and the file structures of tabular form processing system based on graph grammar. (Section 3)

3. We determine file formats of the system. (Section 4)

2. Known Results

[ICSE2001 First Half]

2.1 Modular Tables : Example Hiform [Sugita, 1998]

- Hiform includes all items in ISO6592
- Hiform consists of 17 types of forms

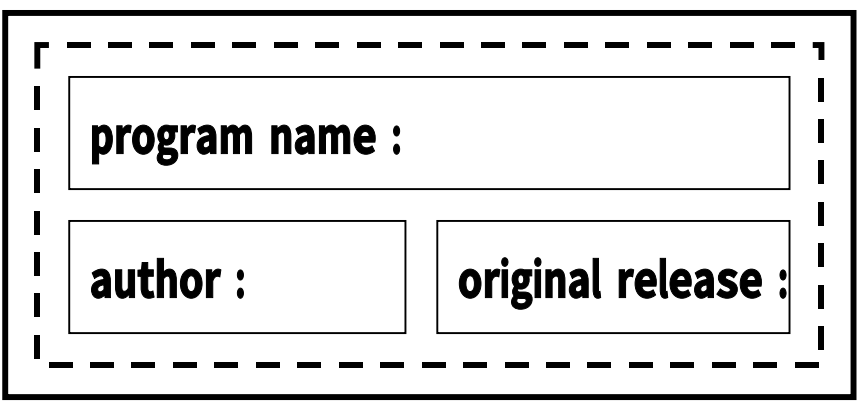
Program Name:	
Subtitle:	
Library Code:	Version:
Author:	Original Release:
Approver:	Current Release:
Problem Description:	
Problem Supplementary Information (Theoretical Principles, Methods and References):	
Problem Solution: 1.Conventions and Terminology 2.Principles and Algorithms	

Program Specification Form

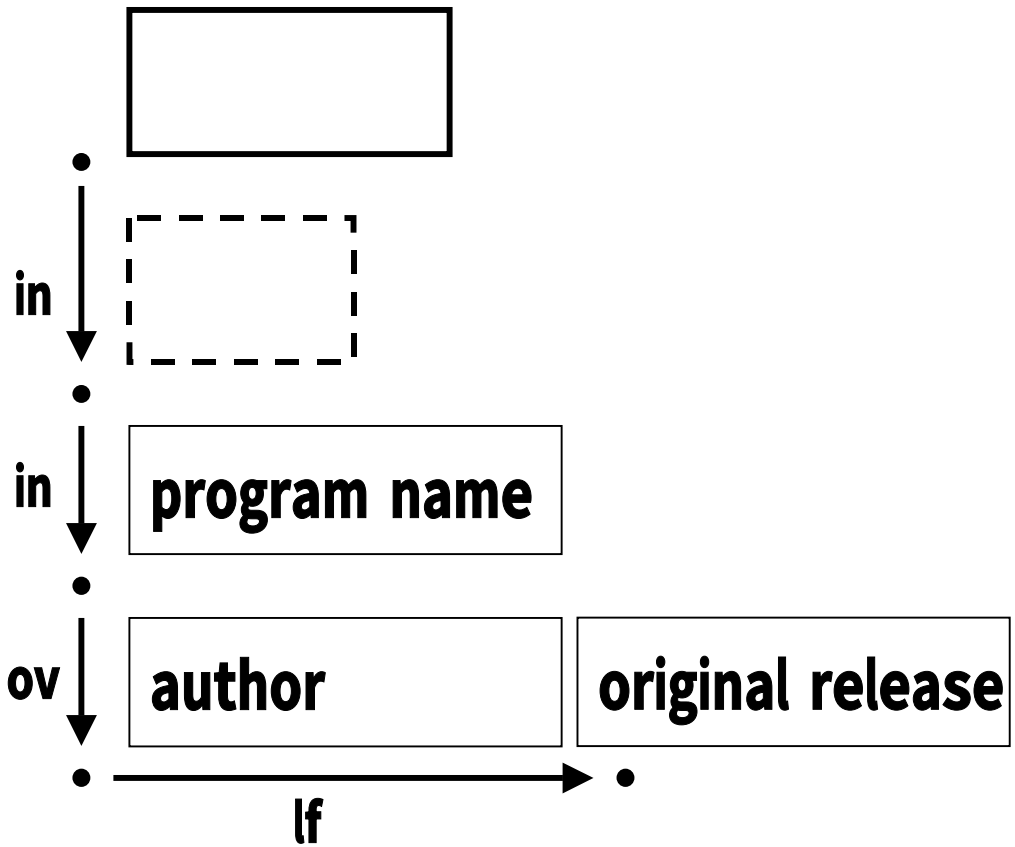
Tabular Forms and Marked Graph [Arita, 2000]

program name :	
author :	original release :

Tabular form



Nested Diagram



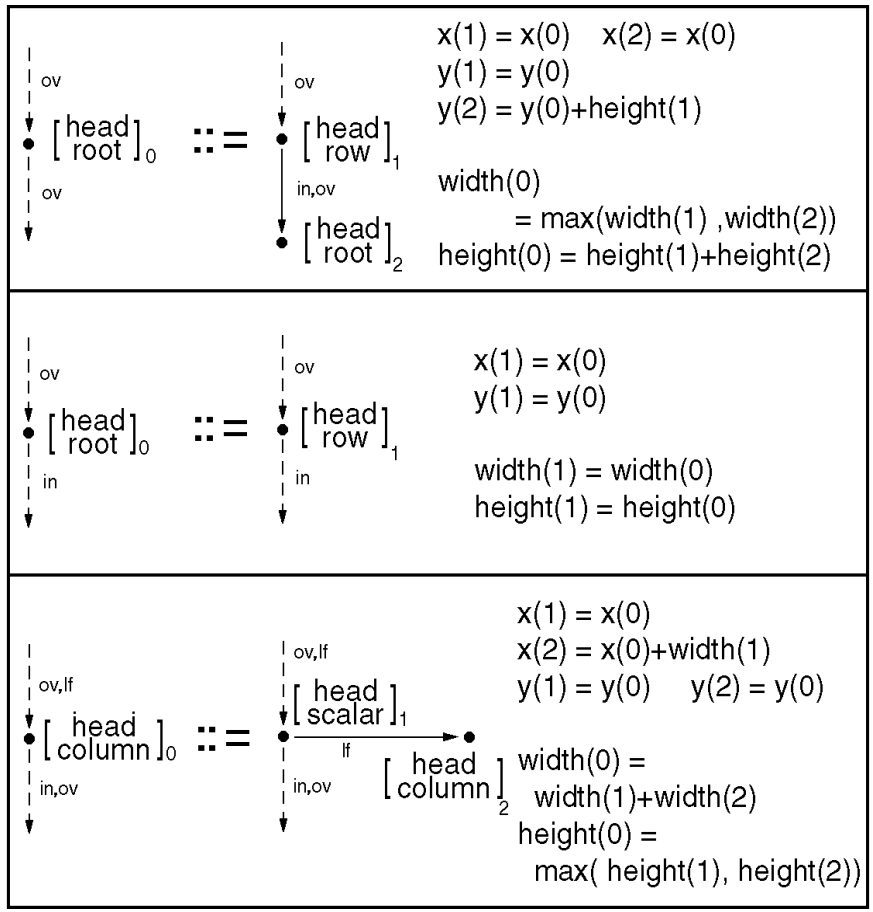
Marked Graph

Attribute edNCE Graph Grammar for Hiform [Arita, 2001]

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

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

P_N : 280 Productions
A_N : The Attributes
F_N : 1248 Attribute rules



A Part of Productions in HNGG

2.2 Parsing Engine (Arita)

Input

Marked Graph with Attribute

MGC(Marked Graph Class)

Output

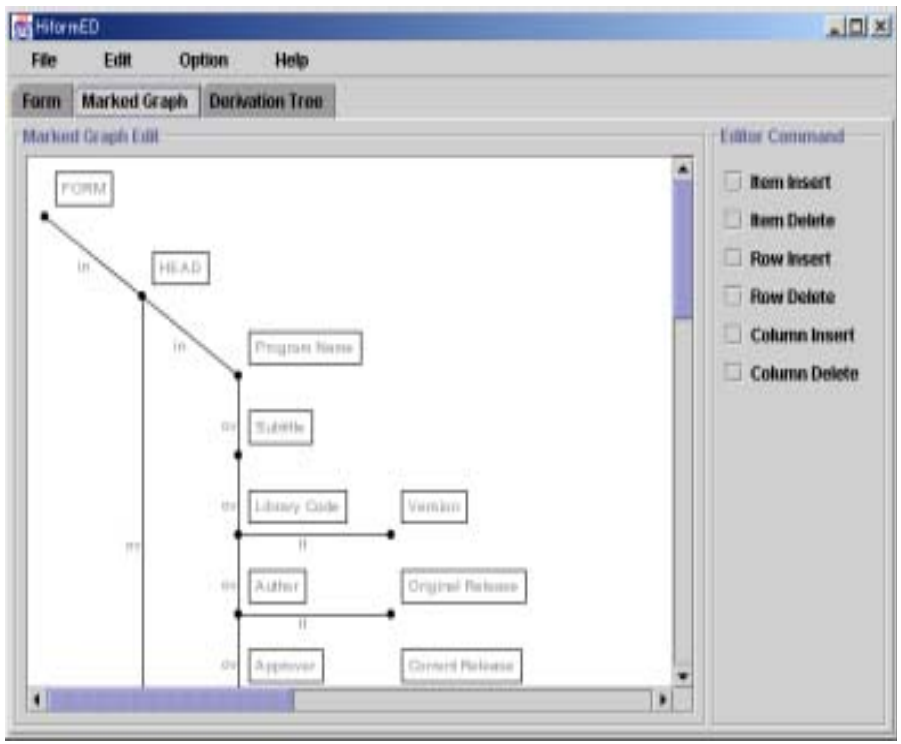
Attribute Derivation Tree

DTC(Derivation Tree Class)

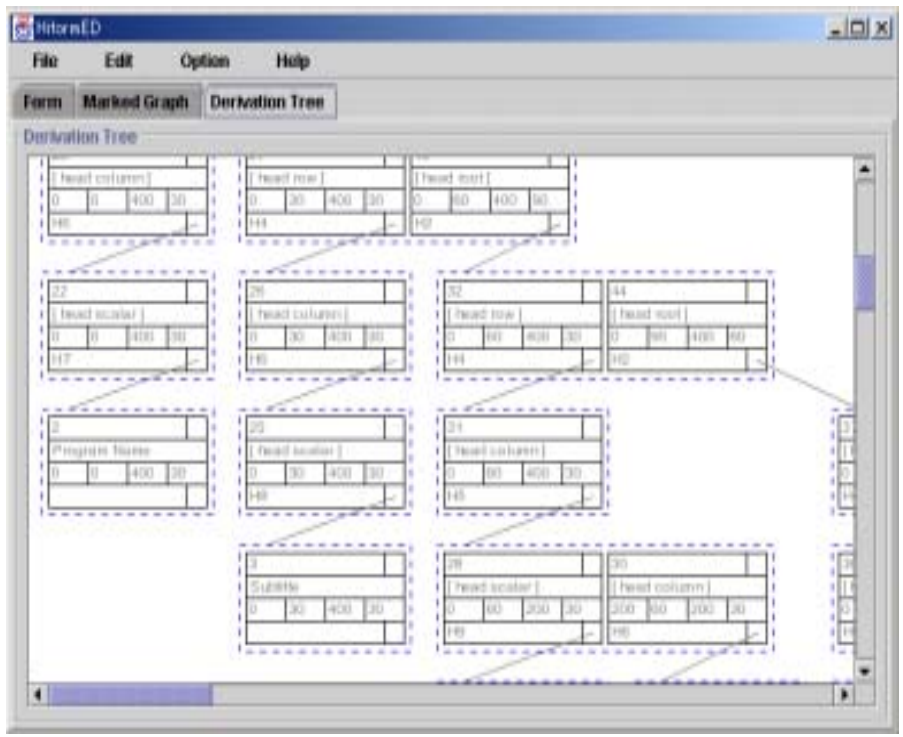
Method

- Syntax Analysis
- Attribute Evaluation

An Execution Screen of Parsing Engine (Arita 2001)



Input : Marked Graph



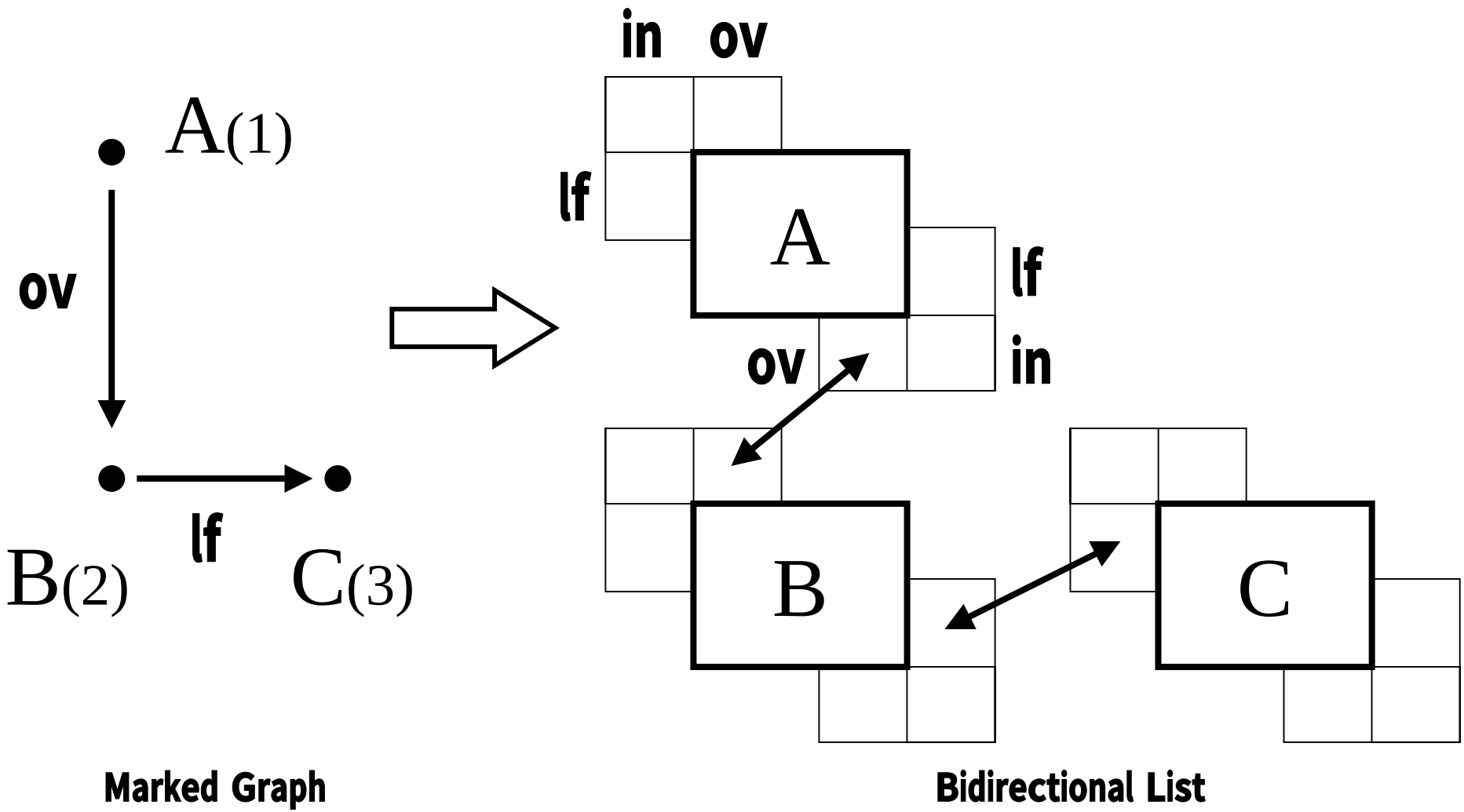
Output : Derivation Tree

2.3 Classes for Marked Graph and Derivation Tree (Arita 2001)

MGC

- Data structure of marked graph (edNCE GG)
- Java Class
- Bidirectional List
- Used for Syntax Analysis

Example : An Illustration of MGC

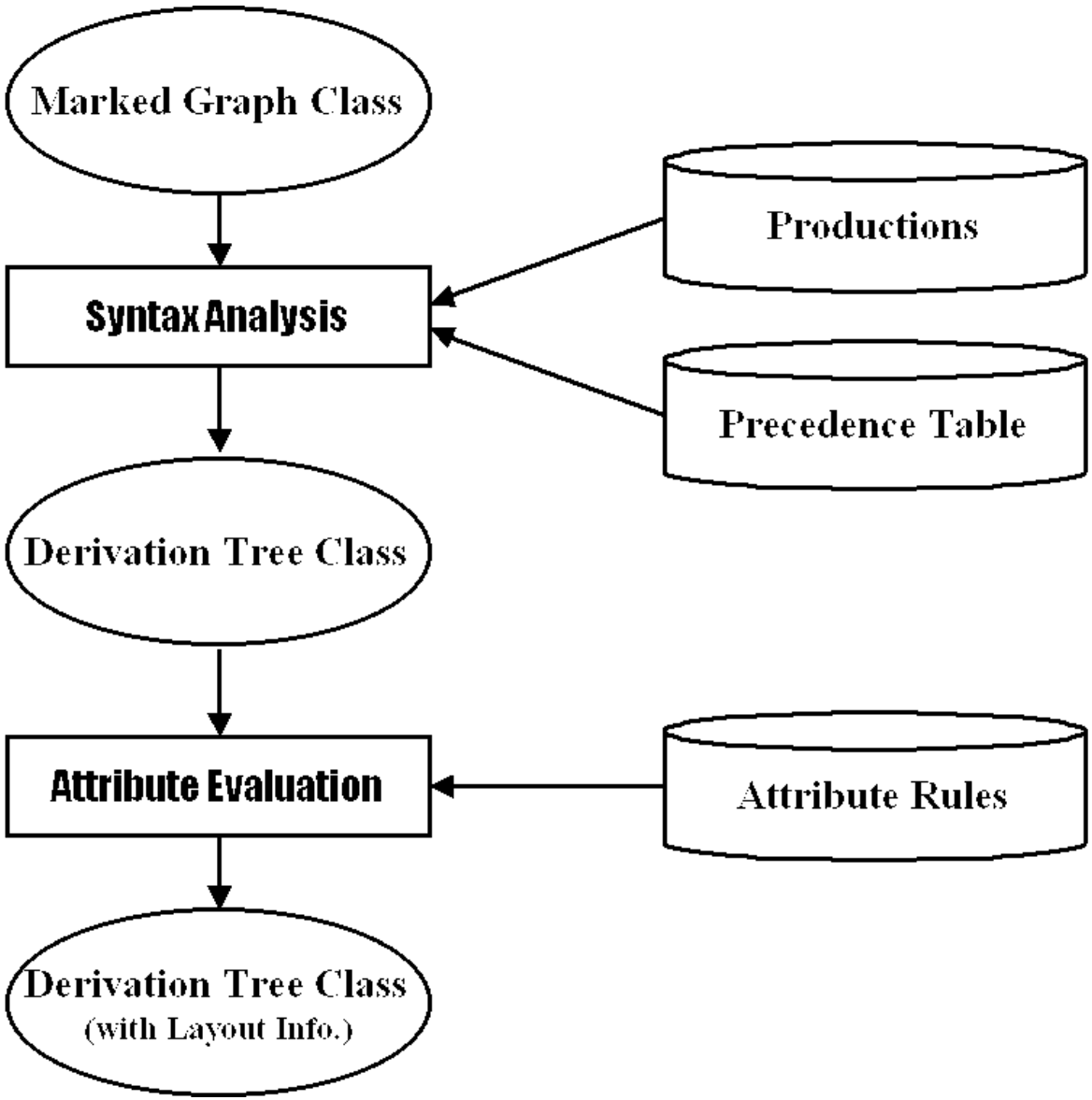


2.3 Classes for Marked Graph and Derivation Tree (Arita 2001)

DTC

- Data structure for derivation tree
- Java Class
- Bidirectional List
- Generated by Syntax Analysis
- Used for attribute evaluation and drawing of a table

Parsing Process



Editing Mechanism of Tabular Forms (Tomiyama)

Feature of Editing Mechanisms

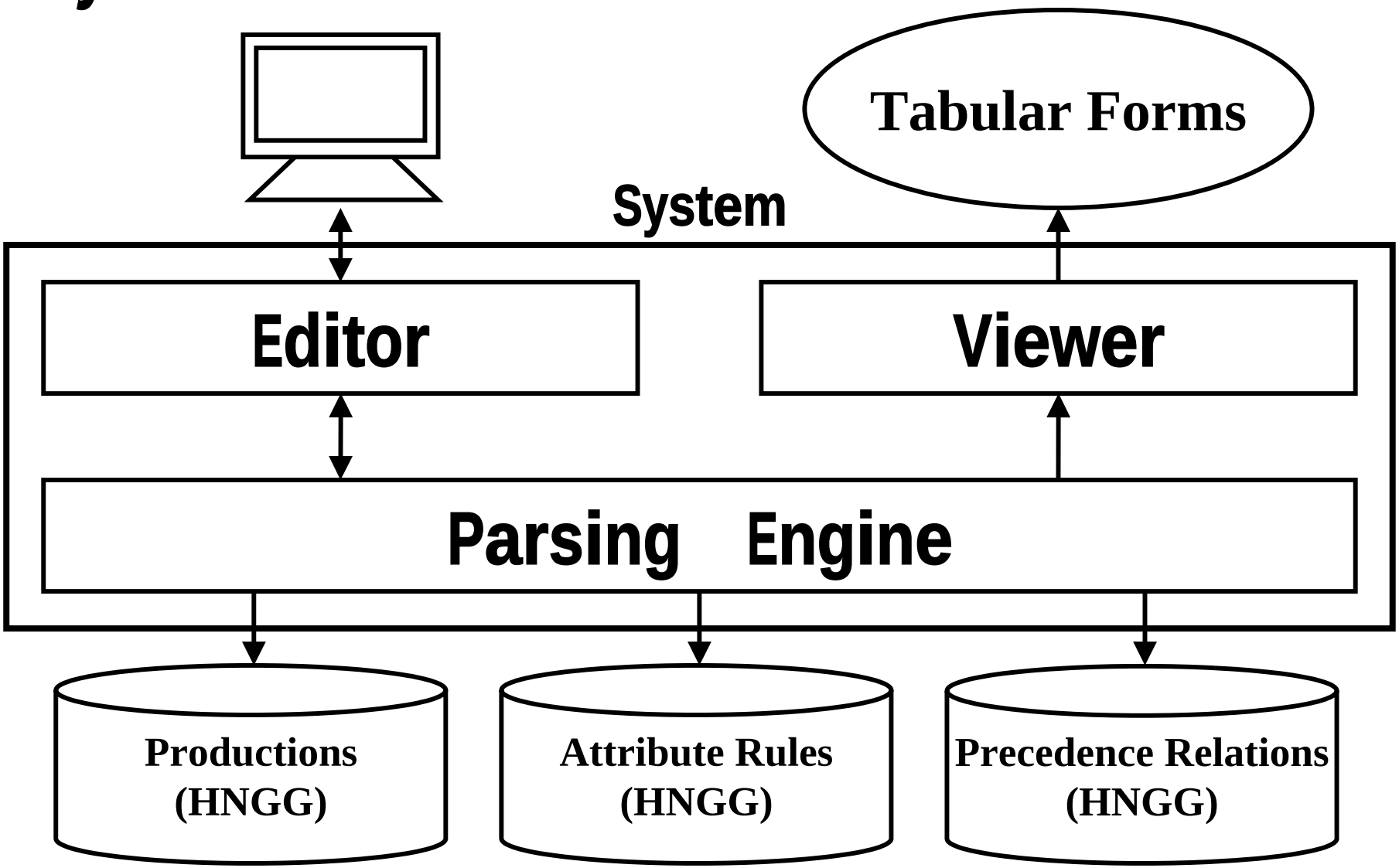
- Formalized mathematically
- Not to cause syntax error
- To include Insertion, Deletion, and Addition

3. System Structure

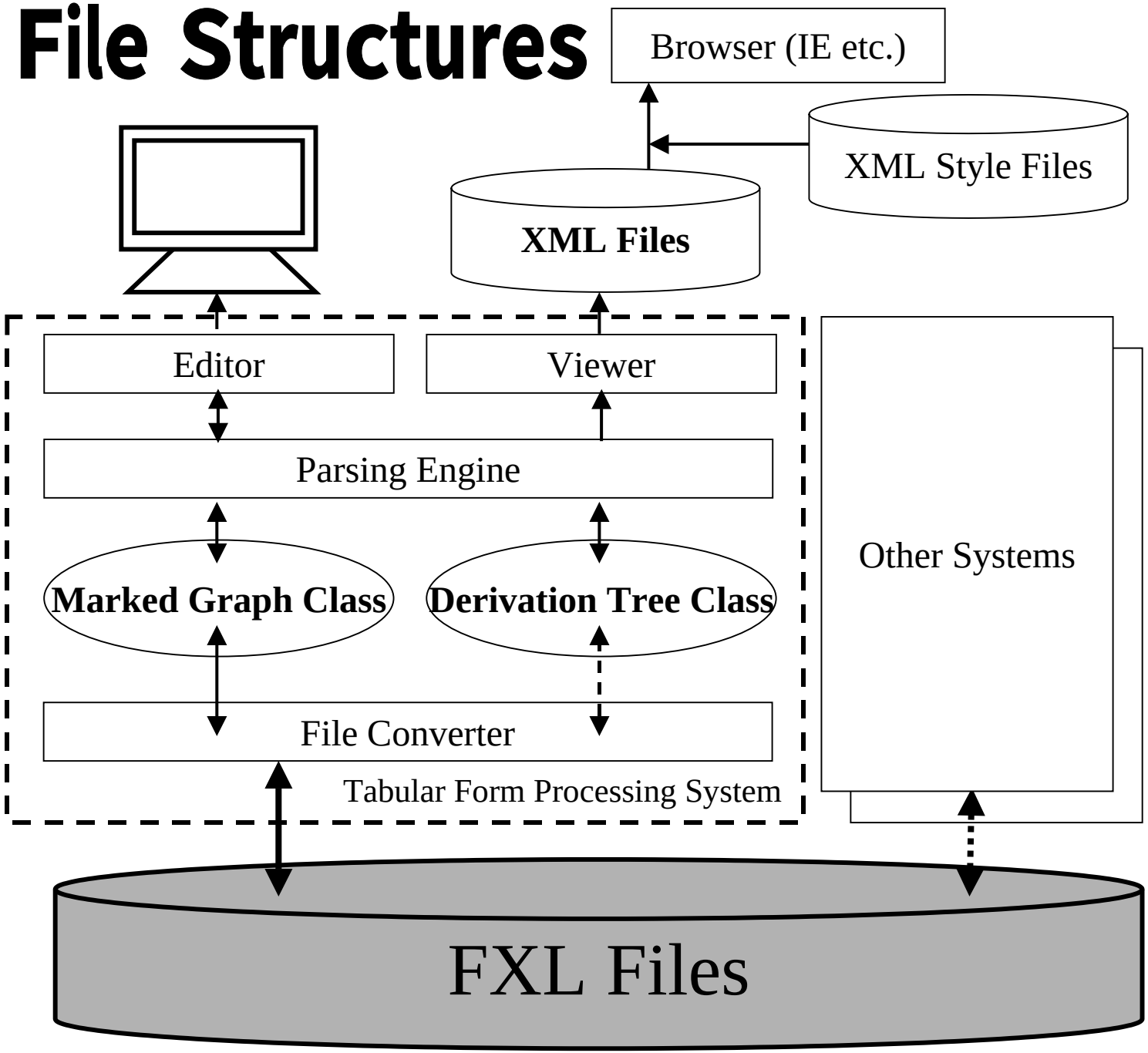
[ICSE2001 Second Half]

3.1 System structure

System Overview



3.2 File Structures



Viewer

Input

DTC(Derivation Tree Class) :

Output of Parsing Engine

Output

XML Files

Other modules omit.

4. File Format

[SPA '02]

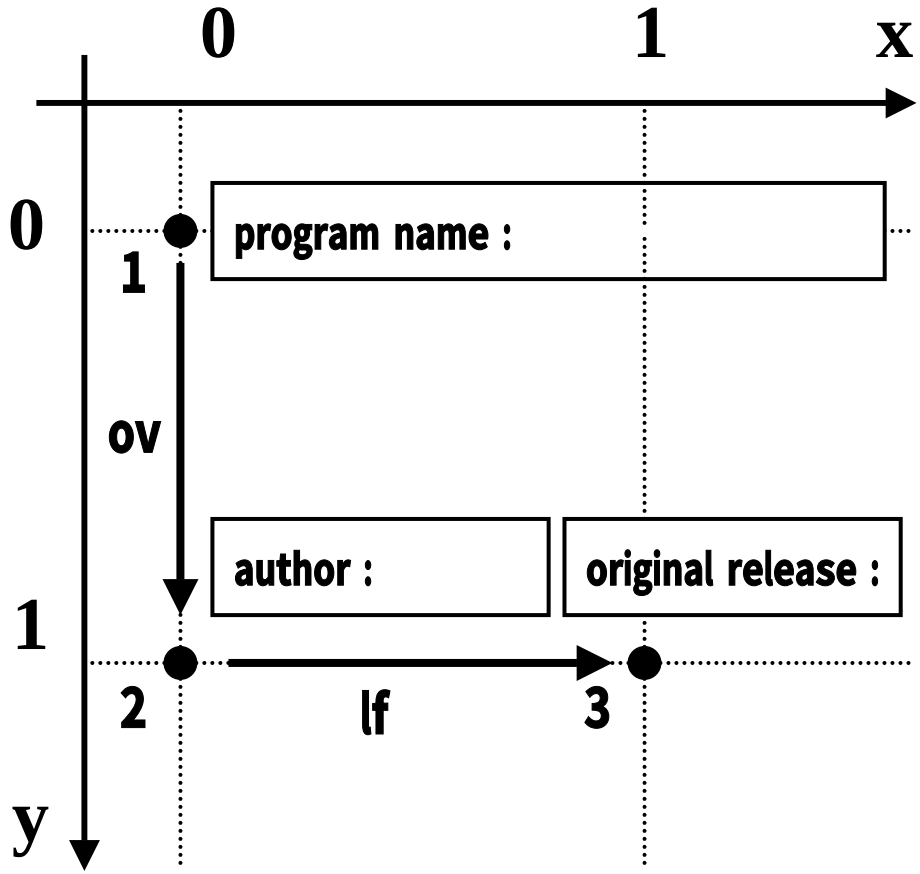
4.1 FXL(A Form eXchange Language)

- Syntax of FXL is defined by extended BNF.
- Codes of FXL are text-based codes.
- FXL can describe several attributes for tabular forms.

4.2 Description of a Graph Part

Whole Structure

```
graph{
    graphHeader{
    date{ 2002,1,1 }
    time{ 0,0,0 }
    }
   nodeSet{nodeObject{'''}
    nodeObject{'''}'''}
   edgeSet{edgeObject{'''}
    edgeObject{'''}''' }
}
```

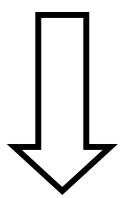


A Graph An Example for FXL Description

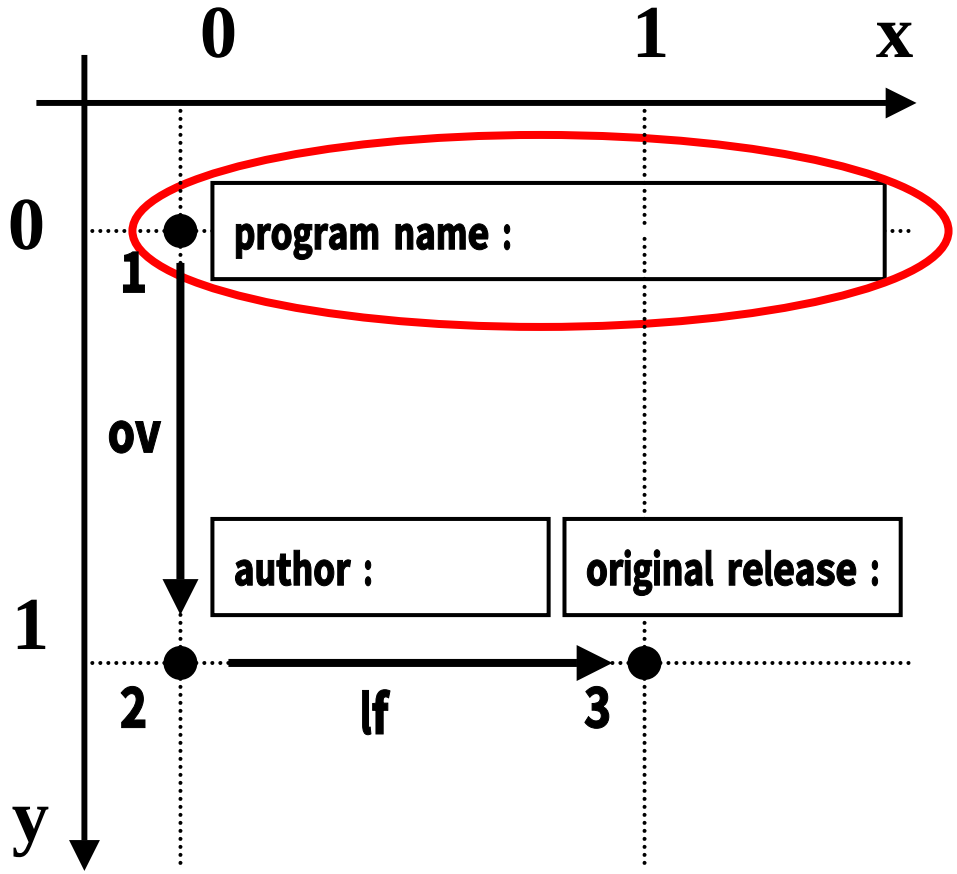
4.3 Description of a Node Part(1)

Description of Node 1

```
nodeObject{
    node{
      nodeID{ 1 }
      nodeX{ 0 }
      nodeY{ 0 }
    }
    nodeLabel{
      labelString{“program name”}
    }
```



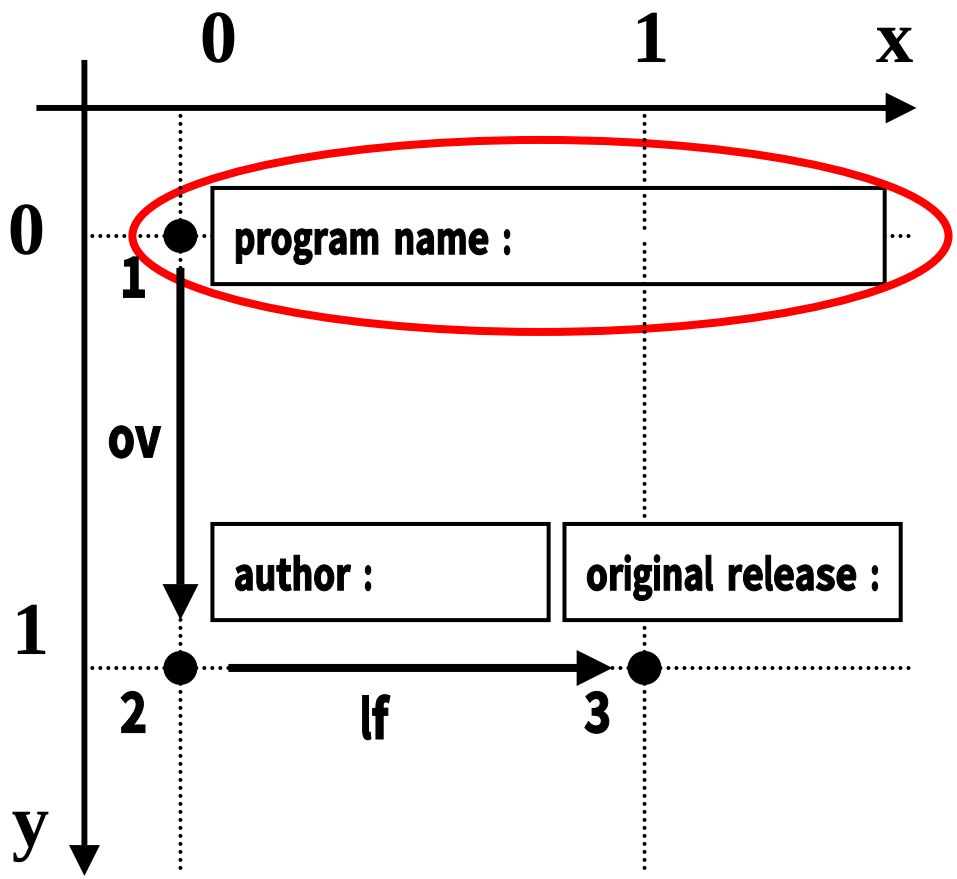
continued



A Graph An Example for FXL Description

4.3 Description of a Node Part(2)

```
attribute{
    cellSize{
      cellWidth{ 2 }
      cellHeight{ 1 }
    }
    cellLocation{
      cellX{ 0 }
      cellY{ 0 }
    }
  }
  cellColor{
    fontRGB{ 0, 0, 0 }
  }
}
```

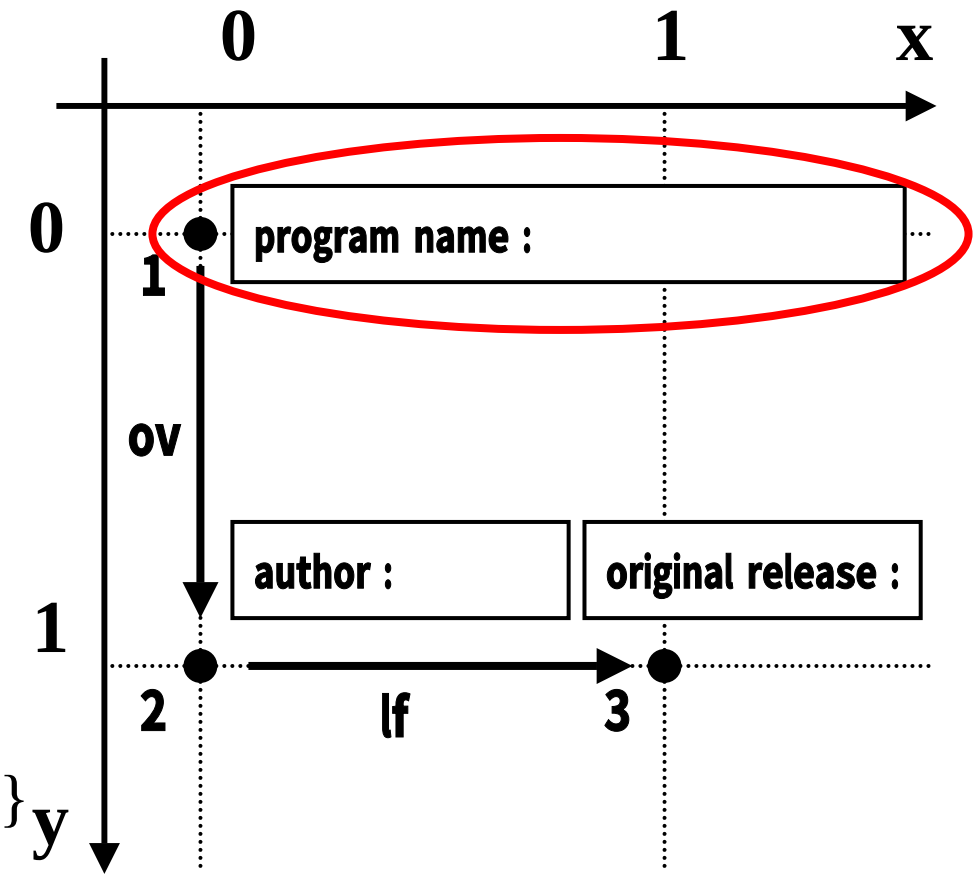


A Graph An Example for FXL Description

4.3 Description of a Node Part(3)

Node and Label Part

```
node{
   nodeID{ 1 }
   nodeX{ 0 }
   nodeY{ 0 }
}
nodeLabel{
   labelString{“program name”}
}
```

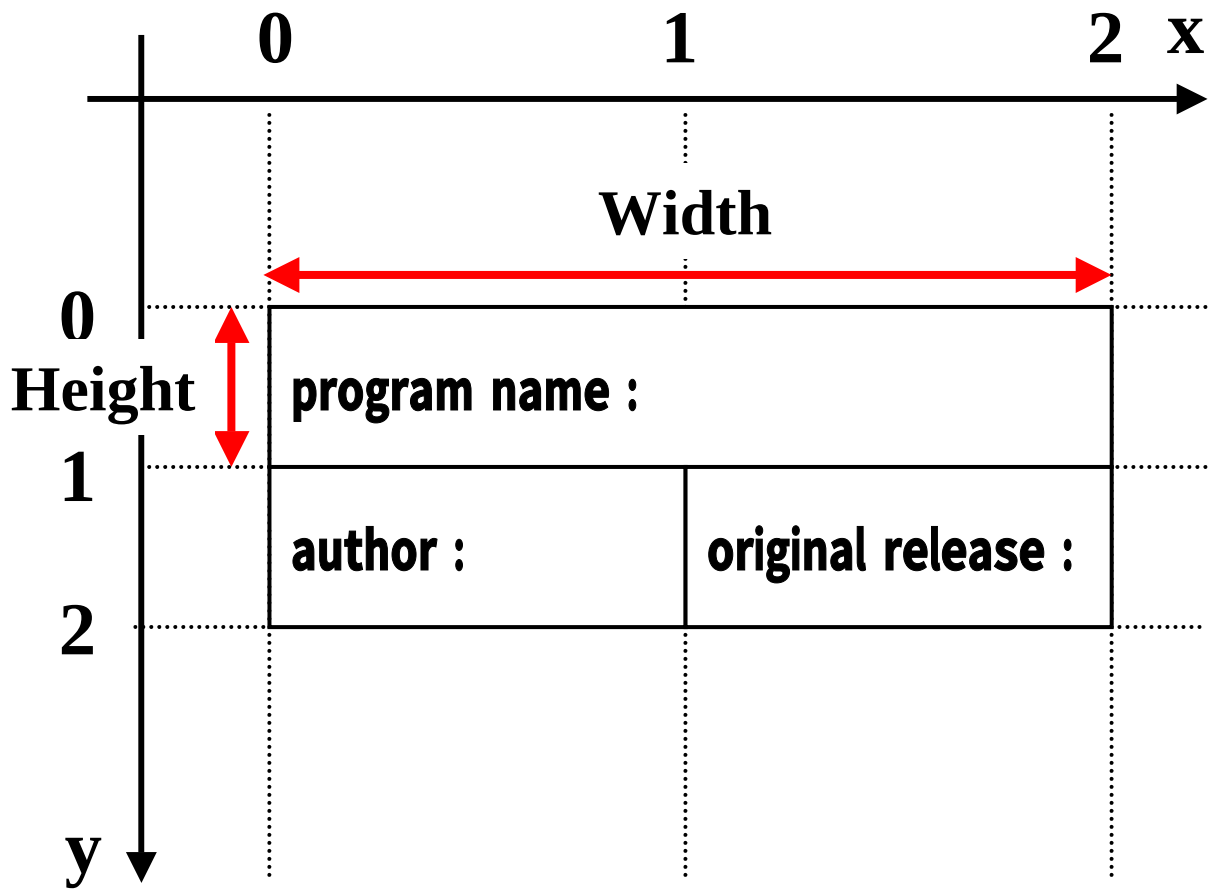


A Graph An Example for FXL Description

4.3 Description of a Node Part(4)

Attribute Part (Layout Information)

```
attribute{
    cellSize{
      cellWidth{ 2 }
      cellHeight{ 1 }
    }
    cellLocation{
      cellX{ 0 }
      cellY{ 0 }
    }
}
```



A Table of the FXL Description

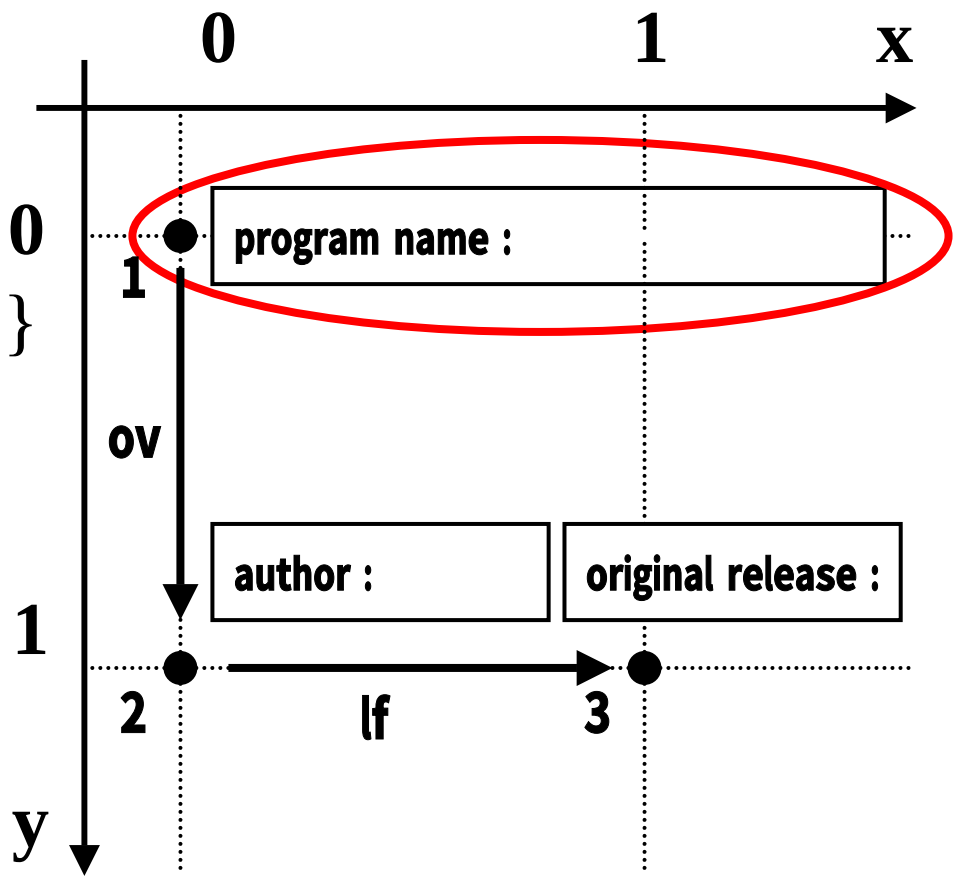
4.3 Description of a Node Part(5)

Color Part

```
cellColor{
  fontRGB{ 255, 255, 255 }
}
```

Color of Cell and Edge

RGB, CMYK, HSL, Gray Scale

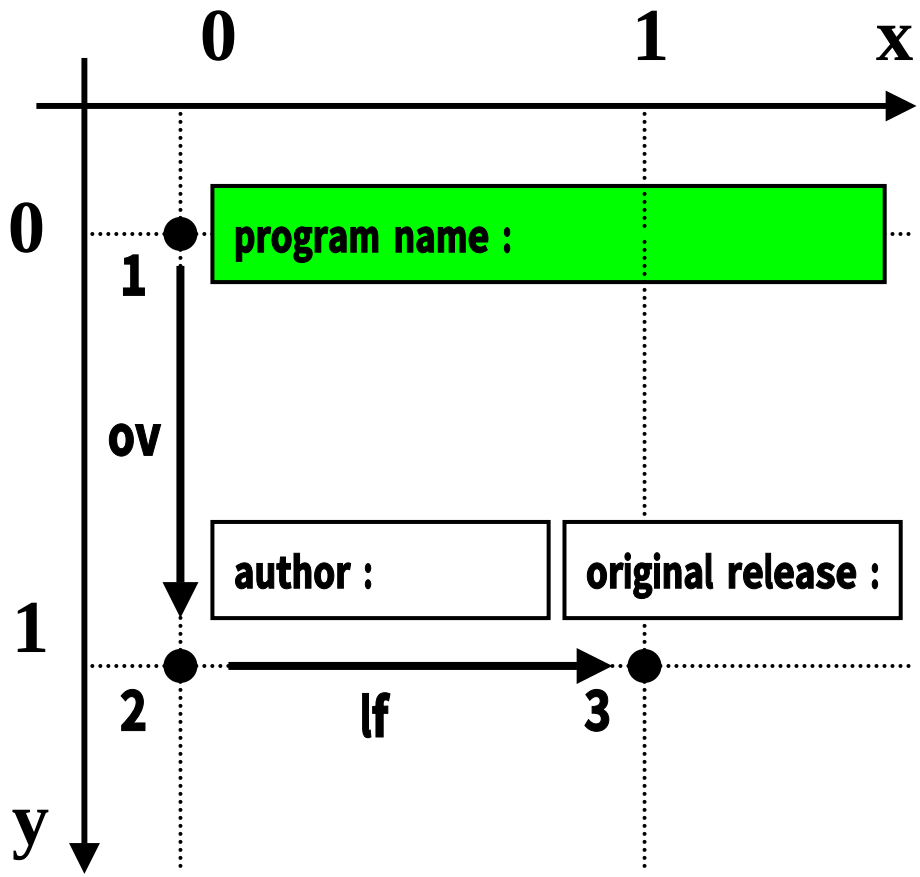


A Graph An Example for FXL Description

An Example of Edge Color

Color Part

```
cellColor{
  fontRGB{ 0, 255, 0 }
}
```

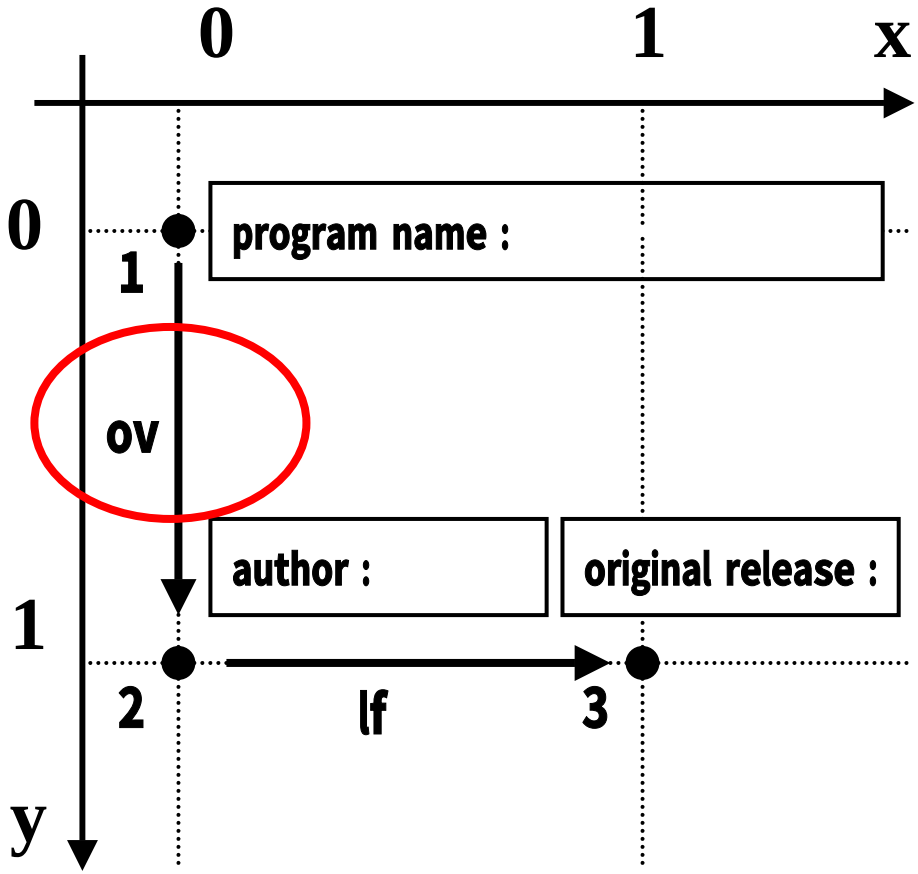


A Graph An Example for FXL Description

4.4 Description of an Edge Part(1)

Description of Edge 1

```
edgeObject{
    edge{
      edgeID{ 1 }
      startNode{ 1 }
      endNode{ 2 }
      edgeShapes{“arrow”}
    }
    edgeLabel{
      labelString{“ov”}
    }
    edgeColor{
      fontCMYK{ 0, 0, 0, 100 }
    }
```

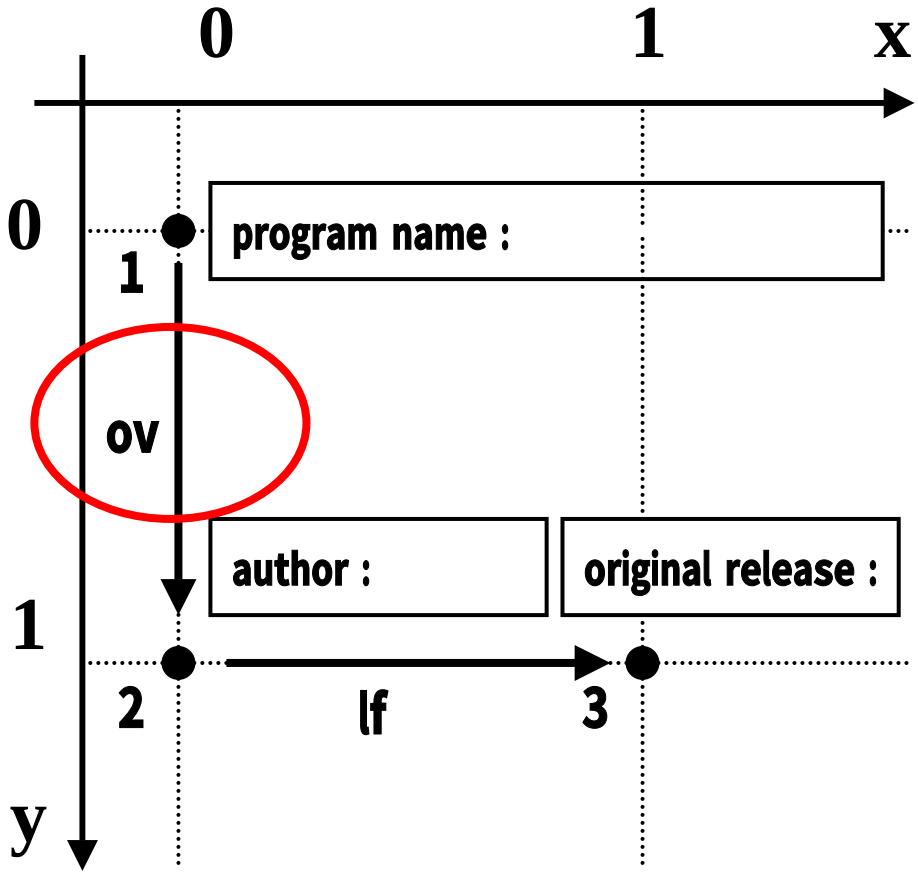


A Graph An Example for FXL Description

4.4 Description of an Edge Part(2)

Edge Part

```
edge{
  edgeID{ 1 }
  startNode{ 1 }
  endNode{ 2 }
  edgeShapes{“arrow”}
}
```

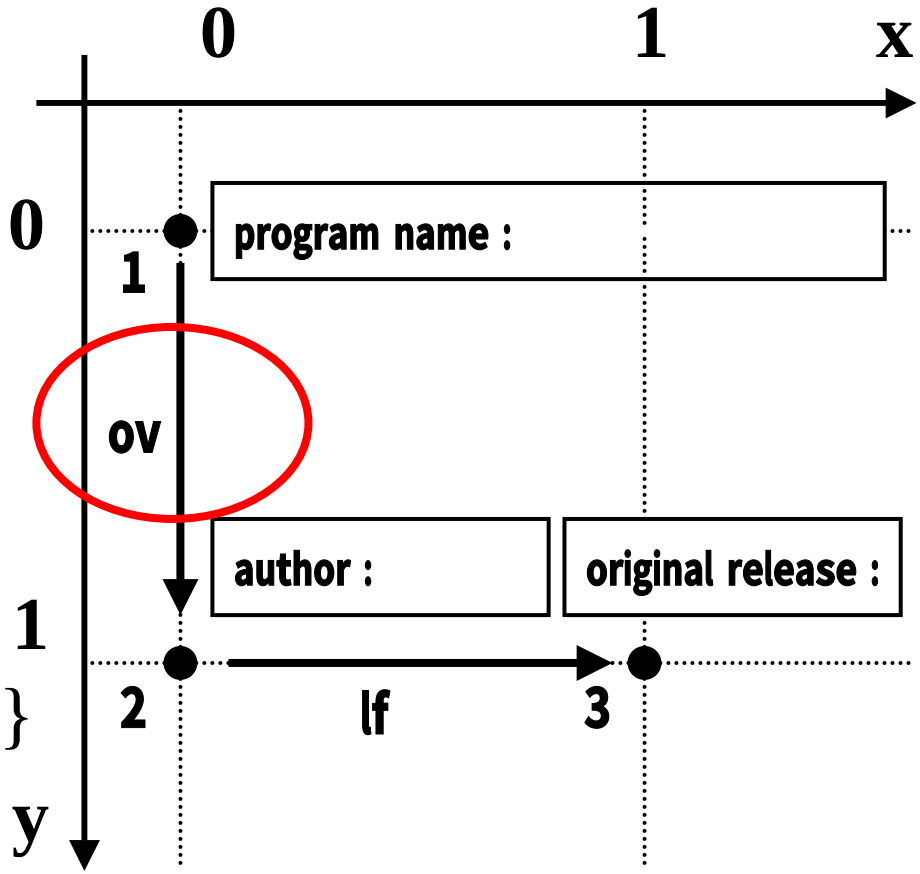


A Graph An Example for FXL Description

4.4 Description of an Edge Part(3)

Label and Color Part

```
edgeLabel{
  labelString{“ov”}
}
edgeColor{
  fontCMYK{ 0, 0, 0, 100 }
}
```

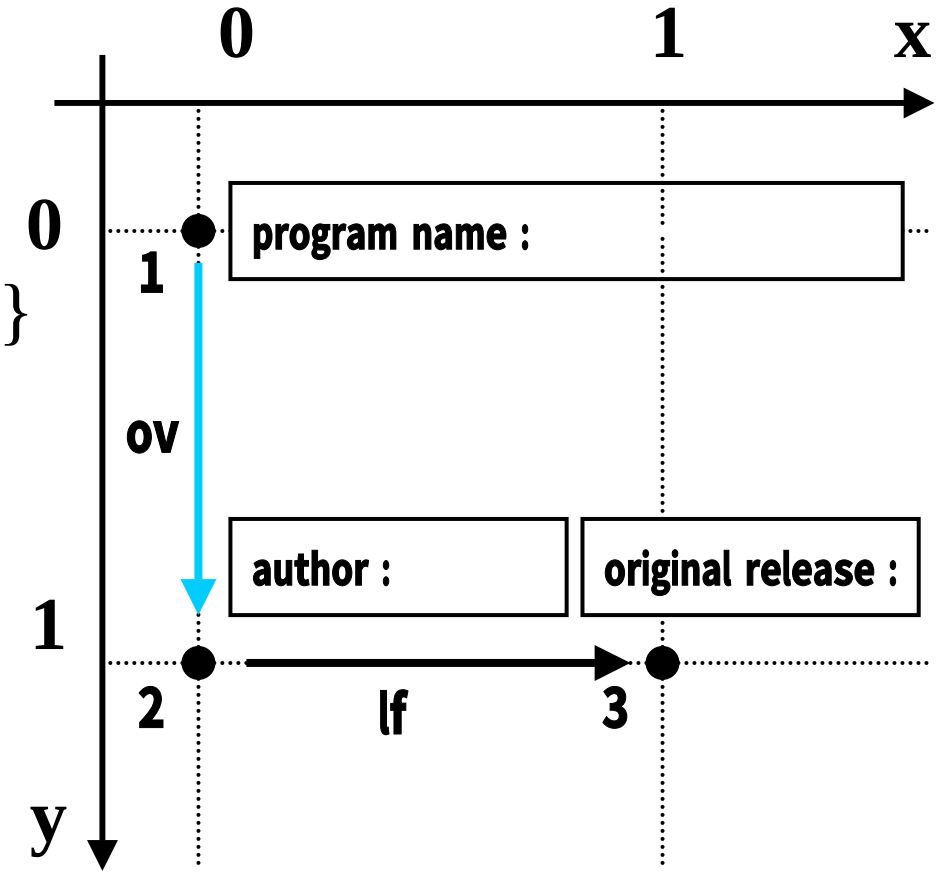


A Graph An Example for FXL Description

An Example of Edge Color

Color Part

```
edgeColor{
  fontCMYK{ 100, 0, 0, 0 }
}
```



A Graph An Example for FXL Description

4.5 BNF for FXL

A Part of Productions in FXL

node_id → "nodeID{" integer '}'
node_position → ε ¦ node_x node_y
node_x → "nodeX{" integer '}'
node_y → "nodeY{" integer '}'

attribute → "attribute{" attribute_inner '}'
attribute_inner → cell_size cell_location cell_color
cell_size → "cellSize{" cell_size_inner '}'
cell_size_inner → cell_width cell_height

Productions : 61 Rules

4.6 Remarks

Remark 1

Type of BNF-FXL is undecided.

Remark 2

The class of marked graphs described by FXL is proper super set of the one by HNGG.

5. Conclusion

5. Conclusion

- We provide short summary, illustrations and examples related.

- We made the system structure and the file structure of tabular form processing system based on graph grammar.

- We determine a file format FXL.

Future Works

- We are planning the development of parsing engine, editor, and viewer based on this file format FXL.

Papers

- A Syntax Directed Environment for Tabular Form Designing, Intarnat. Conf. Software. Engine. ICSE2001
- A Syntax Directed Environment for Tabular Form Processing, 電子情報通信学会全国大会, IEICE2001

  Nakagawa, Arita, Tomiyama, Miyadera, Tsuchida, Yaku

- A XML Viewer for Tabular Forms for use with Mechanical Documentation, Internat. Assoc. Tech. Development. Apply Informatics, IASTED2002

  Inoue, Nakagawa, Arita, Yaku, Tsuchida