数理モデル化と問題解決　40－7
（２００２．６．２６）

拡張論理プログラムにおける推論過程を表す同心円図

笹倉 万里子
sasakura@momo.it.okayama-u.ac.jp
岡山大学 工学部 情報工学科

概要

本稿では拡張論理プログラムにおける推論過程を視覚化するための新しい方法を提案する。拡張論理プログラムにおける推論過程は再帰的な構造をしているので、再帰構造を直観的に表せる同心円を使う。従来の推論手続きに視覚化のためのステップを追加した視覚化用推論手続きを構築し、それによって生成された図を例示する。

Concentric Circle Diagrams for Visualizing a Reasoning Process on an Extended Logic Program

Mariko Sasakura
Department of Information Technology, Okayama University

Abstract

We propose a new visualization for a reasoning process on an extended logic program. We use concentric circles to visualize recursive structures in a reasoning process. We add a process for our visualization to an existing reasoning process. We explain the details of our procedure and show some example diagrams.

1 Introduction

Studies on visualization are classified broadly into scientific visualization and information visualization. Scientific visualization is for seeing data obtained by experiments, observations or simulations on such as fluid mechanics, atmospheric phenomena, medical aid and genome analysis. Information visualization is for seeing abstract information such as databases, webs or programs[1].

Our study described in this paper is one of information visualizations. We visualize a reasoning process on an extended logic program to help users understand how the process is going intuitively.

Reasoning procedures on extended logic programs are already proposed and their semantics are discussed in [2], [4]. Our approach is to add a visualization function to an existing procedure.

At first, we give an intuitive explanation of our approach in section 2. Then, we briefly explain our reasoning and visualization procedure, named an abductive procedure for Concentric Circle Diagrams, in section 3. In section 4 we show figures generated by our procedure and we give conclusion in section 5.

2 Intuitive explanation

In extended logic programs, there are two kinds of negation: a strong negation $\neg l$ and a negation as failure $\sim l$, where l is a literal. Reasonings are performed under the following constraints[2],[4].

- Both l and $\neg l$ do not succeed at a time.
- Both l and $\sim l$ do not succeed at a time.
- If $\neg l$, then $\sim l$.
- If $\sim l$ succeeds, then $\neg l$ succeeds.

A reasoning process forms a recursive structure. Therefore, we visualize the reasoning process as concentric circles that is suitable to represent recursive structures.

For example, there is an extended logic program:

$$p \leftarrow q$$
$$q \leftarrow r$$
$$r \leftarrow$$

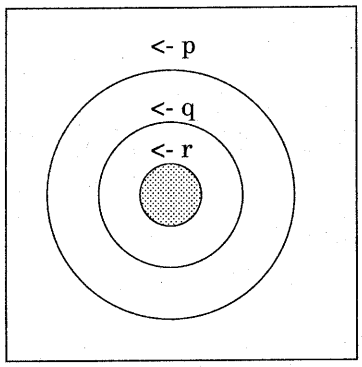

Figure 1: An example of our visualization for a reasoning process

If we give p as an initial goal, a reasoning process finds p succeeds, since q succeeds because of r success. This process is visualized by our procedure as figure 1. The detail of the procedure will be explained in the next section.

3 An abductive procedure for Concentric Circle Diagrams

An abductive procedure for Concentric Circle Diagrams consists of two derivations: a succeeding derivation and a failing derivation. A succeeding derivation tries to find a given goal succeeds. A failing derivation tries to find a given goal fails. A succeeding derivation and a failing derivation invokes each other if necessary. An abductive procedure for Concentric Circle Diagrams we describe in this paper is based on the procedure shown in [5]. We add elements for visualization to the procedure and rearrange it.

Now, we explain our succeeding derivation and failing derivation. A given extended logic program is denoted as P.

A succeeding derivation is expressed as the following list:

$$(G_0, \Sigma_0, DN_0), \ldots, (G_h, \Sigma_h, DN_h).$$

This is also represented as the following:

$$(G_0, \Sigma_0, DN_0) \Rightarrow_{suc} (G_h, \Sigma_h, DN_h).$$

Assume a goal is $G = M_1, \ldots, M_m$, where M_i is a literal or a negation as failure of a literal. Σ is a set of literals that succeed in this derivation. DN is information for visualization of which structure is:

$$DN_i = (D_i, \Delta_i, DL_i)$$
$$D_i = C_0, \ldots, C_i$$
$$DL_i = DN_{i_0}, \ldots, DN_{i_n}$$

where, $C_i = (G_i, A_i)$. We call A_i as an attribute of which value is *start*, *rewrite* or *call*. The initial value of C is $C_0 = (G_0, start)$.

If a succeeding derivation halts and $G_h = \Box$, then we call G_0 succeeds.

A failing derivation succeeds when a given set of goals fail. We must find all goals generated by the given goals fail for success of a failing derivation. A failing derivation is expressed as

$$(H_0, \Sigma_0, DNS_0), \ldots, (H_h, \Sigma_h, DNS_h).$$

This is also represented as

$$(H_0, \Sigma_0, DNS_0) \Rightarrow_{ff} (H_h, \Sigma_h, DNS_h).$$

H_i is a set of goals and DNS_i is a set of DNs. When a failing derivation halts and $H_h = \emptyset$ then H_0 succeeds. Σ is a set of literals that succeed in this derivation.

The details of the derivations that are how we get $(G_{n+1}, \Sigma_{n+1}, DN_{n+1})$ from (G_n, Σ_n, DN_n) and $(H_{n+1}, \Sigma_{n+1}, DNS_{n+1})$ from (H_n, Σ_n, DNS_n), are explained in [3].

We show an example of which goal is A and program P_1 that is:

$$A \leftarrow\sim B$$
$$B \leftarrow C$$

At first, a succeeding derivation is invoked for $G_0 = \leftarrow A$. By the first clause of P_1, $G_1 = \leftarrow\sim B$. Then a failing derivation is invoked for $H_0 = \{\leftarrow B\}$. By the second clause of P_1, $H_1 = \{\leftarrow C\}$. Since there is no clause whose head is C, $H_2 = \emptyset$. Therefore the failing derivation succeeds. Then the succeeding derivation succeeds. Table 1 shows G, H, Σ and DN of the derivations.

4 Drawing

In this section, we explain how to draw diagrams from DN of a succeeding derivation and DNS of a failing derivation.

Table 1: A reasoning proccess on program P_1 for $G_0 = \leftarrow A$

A succeeding derivation for $\leftarrow A$

G	Σ	DN
$G_0 = \leftarrow A$	$\emptyset$	$(((\leftarrow A, start)), \emptyset, NULL)$
$G_1 = \leftarrow\sim B$	$\{A\}$	$(((\leftarrow A, start),$ $(\leftarrow\sim B, rewrite)), \emptyset, NULL)$
$G_2 = \Box$	$\{A\}$	$(((\leftarrow A, start),$ $(\leftarrow\sim B, rewrite),$ $(\Box, call)), \{B\}, DN_f)$

A failing derivation for $\{\leftarrow B\}$

H	Σ	DNS
$H_0 = \{\leftarrow B\}$	$\{A\}$	$\{(((\leftarrow B, start)), \{B\}, NULL)\}$
$H_1 = \{\leftarrow C\}$	$\{A\}$	$\{(((\leftarrow B, start),$ $(\leftarrow C, rewrite)), \{B\}, NULL)\}$
$H_2 = \{\emptyset\}$	$\{A\}$	$\{(((\leftarrow B, start),$ $(\leftarrow C, rewrite),$ $(\emptyset, rewrite)), \{B\}, NULL)\}$

4.1 Overview

When we execute an abductive procedure for Concentric Circle Diagrams, we get $(G_0, \Sigma_0, DN_0), \Rightarrow_{suc} (G_h, \Sigma_h, DN_h)$ for a given goal. The DN_h records all goals of succeeding and failing derivations invoked by the procedure. We visualize the DN_h as a process of a reasoning.

The transformation from a DN_h to diagrams is a one to one function from an element of DN_h to an element of a diagram. We describe it in the next subsection.

4.2 How to draw CCDs

DN_h consists of a list of D that is a list of C_i which is (G_i, A_i), Δ and a list of DN. We draw a diagram for a D. The correspondence of an element of D and an element of a diagram is the following:

D	a diagram
C_i	a circle
G_i	a label
A_i	colour of a circle

A diagrams is an area surrounded by a rectangle as figure 1. We draw a background of a diagram for a failing derivation as gray so that we can easily know which diagram shows a succeeding derivation and which one shows a failing derivation. A circle corresponds to C_i is drawn in the area and put with a label that corresponds to G_i. The colour of the circle is decided by A_i. But, for the innermost

begin
  if $G_i = \Box$ in a succeeding derivation
    then a colour of a circle is gray.
  else if $H_i = \emptyset$ in a failing derivation
    then a colour of a circle is black.
  else if $A_i = call$
    then a colour of a circle is dark gray.
  else a colour of a circle is white.
end

Figure 2: A colour of a circle

circle, the colour is decided by G_i. Figure 2 shows how to decide a colour of a circle.

If r_i is a radius of a circle for C_i and r_j is a radius of a circle for C_j and $i < j$ then $r_i > r_j$. The centers of all circles in a diagram have the same coordinates. Therefore, we show a diagram consists of concentric circles and smaller circles denote latter goals.

We draw a diagram for a DN. So, generally, a reasoning process for a goal is represented by several diagrams.

4.3 Examples

Figure 3 shows a reasoning process of the following program P_2 for $\leftarrow p$.

$$p \leftarrow\sim q, r$$
$$q \leftarrow s$$
$$q \leftarrow \neg t$$
$$r \leftarrow$$

The leftside diagram represents a succeeding derivation for $\leftarrow p$ and two diagrams on the rightside represent failing derivations for $\leftarrow q$. There are two clauses of which head is q in P_2, we must show the failure of q in both clauses. Two diagrams on the rightside correspond to the clauses.

5 Conclusions

We propose an abductive procedure for Concentric Circle Diagrams which visualizes a reasoning process on an extended logic program. We use concentric circles to visualize a reasoning process so that we may see recursive structures in a reasoning process intuitively.

In this paper, we do not mention the soundness of our procedure. We will discuss it in the future paper.

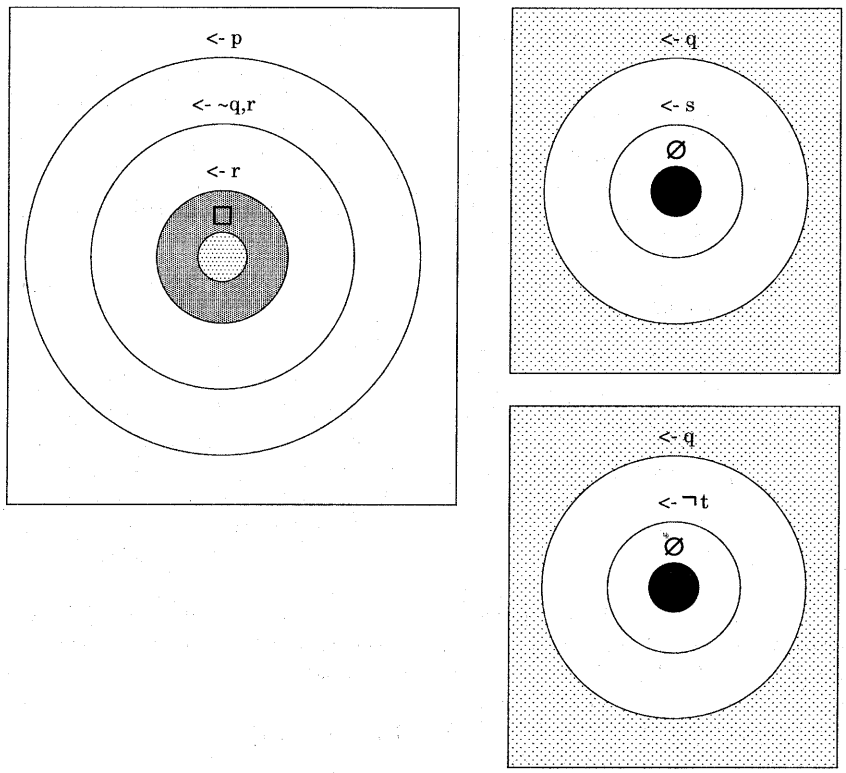

Figure 3: Diagrams of a reasoning process for $\leftarrow p$ on P_2

References

[1] R. Spence. *Information Visualization*, Addison-Wesley, 2001.

[2] R.A.Kowalski and F.Sadri. Logic programs with exceptions, *Proc. of 7th International Conference on Logic Programming*, 598–613, 1990.

[3] M.Sasakura, Concentric Circle Diagrams for Visualizing a Reasoning Process on an Extended Logic Program PDPTA02, 2002.

[4] S.Yamasaki et al., Contradiction-free abduction framework for extended logic program, *Journal of Japanese Society of Artificial Intelligence*, 15(4), 719–726, 2000 (in Japanese).

[5] S.Yamasaki ans M.Sasakura, Towards Distributed Programming Systems with Visualizations Based on Nonmonotonic Reasoning *SSGRR2001(CD-ROM)*, 2001.