数理解析研究所講究録
1093 巻 1999 年 117-122

Refutable Inference of Formal Graph Systems and NLC Graph Grammars

Tetsuhiro Miyahara (宮原 哲浩)† Tomoyuki Uchida (内田 智之)†
Takayoshi Shoudai (正代 隆義)‡ Kenichi Takahashi (高橋 健一)†
Hiroaki Ueda (上田 祐彰)†

†Faculty of Information Sciences, Hiroshima City University
Hiroshima 731-3194, Japan
{miyahara@its,uchida@cs,takahasi@its,ueda@its}.hiroshima-cu.ac.jp
‡Department of Informatics, Kyushu University 39
Kasuga 816-8580, Japan
shoudai@i.kyushu-u.ac.jp

Abstract

A graph is one of the most common abstract structures and is suitable for representing relations between various objects. The analyzing system directly manipulating graphs is useful for knowledge discovery. Formal Graph System (FGS) is a kind of logic programming system which directly deals with graphs just like first order terms. We have designed and implemented a knowledge discovery system KD-FGS, which receives the graph data and produces a hypothesis by using FGS as a knowledge representation language. The system consists of an FGS interpreter and a refutably inductive inference algorithm for FGSs.

In this paper, we show that sufficiently large hypothesis spaces of FGS programs are refutably inferable and thus give a theoretical foundation of KD-FGS. We report some experiments of running KD-FGS and confirm that the system is useful for knowledge discovery from graph data. We also consider refutable inference of NLC graph grammars.

1. Introduction

Machine learning and data mining technology have been used for knowledge discovery and prediction in many fields [4]. The aim of knowledge discovery is to find a small and understandable hypothesis which explains data nicely. A graph is one of the most common abstract structures and is suitable for representing relations between various objects [10]. We believe that the analyzing system directly dealing with graphs is useful for knowledge discovery.

Formal Graph System (FGS, [8]) is a kind of logic programming system which directly deals with graphs just like first order terms. So FGS is suitable to represent logical knowledge explaining given graph data. We have designed and implemented a knowledge discovery system KD-FGS (see Figure 1), which receives graph data and produces a hypothesis by using FGS as a knowledge representation language. As inputs, the system receives positive and negative examples of graph data. As an output, the system produces an FGS program which is consistent with the positive and negative examples if such a hypothesis exists. Otherwise, the system refutes the hypothesis space. KD-FGS consists of an FGS interpreter and a refutably inductive inference algorithm of FGS programs. The FGS interpreter is used to check whether a hypothesis is consistent with the given graph data or not.

A refutably inductive inference algorithm, proposed by Mukouchi and Arikawa [5], is a special type of inductive inference algorithm with refutability of hypothesis spaces. Suppose that a hypothesis space is refutably inferable and data are succes-

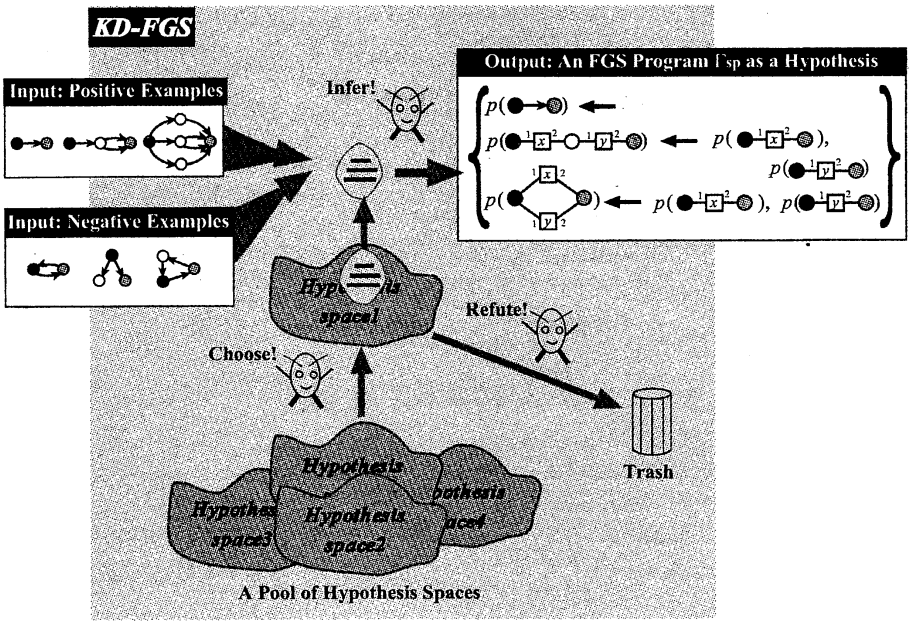

Figure 1: KD-FGS: a knowledge discovery system from graph data using FGS

sively given to the algorithm for the hypothesis space. If there exists a hypothesis describing the data in the hypothesis space, then the algorithm will infer the hypothesis, that is, it will eventually identify the hypothesis. If not, then the algorithm will refute the hypothesis space, that is, it will tell us that no hypothesis in the hypothesis space explains the data and stop. When the hypothesis space is refuted, KD-FGS chooses another hypothesis space and tries to make a discovery in the new hypothesis space. By refuting the hypothesis space, the algorithm gives important suggestions to achieve the goal of knowledge discovery. Thus, KD-FGS is useful for knowledge discovery from graph data. KD-FGS is based on FGS, which is a kind of graph generating systems. We show that the method of KD-FGS is also applicable to the discovery system based on NLC graph grammar, which is a famous graph generating system.

This paper is organized as follows. In Section 2, we introduce FGS as a new knowledge representation language for graph data. In Section 3, we give our framework of refutably inductive inference of FGS programs. For constant $n \geq 0$, we show that the hypothesis spaces of weakly reducing or size-bounded FGS programs with at most n graph rewriting rules are refutably inferable. Thus we give a theoretical foundation of KD-FGS. In Section 4, we report implementation of KD-FGS and some experimental results. In Section 5, we also consider refutable inference of NLC graph grammars.

2. FGS as a New Knowledge Representation Language

Formal Graph System (FGS, [8]) is a kind of logic programming system which directly deals with graphs just like first order terms.

Let Σ and Λ be finite alphabets, and let X be an alphabet, whose element is called a *variable label*. Assume that $(\Sigma \cup \Lambda) \cap X = \emptyset$. A *term graph* $g = (V, E, H)$ consists of a vertex set V, an edge set E and a multi-set H whose element is a list of distinct vertices in V and is called a *variable*. And a term graph g has a vertex labeling $\varphi_g : V \to \Sigma$, an edge labeling $\psi_g : E \to \Lambda$ and a variable labeling $\lambda_g : H \to X$. A term graph is called *ground* if $H = \emptyset$. For example, a term graph $g = (V, E, H)$ is shown in Figure 2, where $V = \{u_1, u_2\}$, $E = \emptyset$, $H = \{e_1 = (u_1, u_2), e_2 = (u_1, u_2)\}$, $\varphi_g(u_1) = s$, $\varphi_g(u_2) = t$, $\lambda_g(e_1) = x$, and $\lambda_g(e_2) = y$. An *atom* is an expression of the form $p(g_1, \ldots, g_n)$, where p is a predicate symbol with arity n and $g_1, \ldots, g_n$ are term graphs. Let $A, B_1, \ldots, B_m$ be atoms with $m \geq 0$. Then, a *graph rewriting rule*

is a clause of the form $A \leftarrow B_1, \ldots, B_m$. An *FGS program* is a finite set of graph rewriting rules. For example, the FGS program Γ_{SP} in Figure 1 generates the family of all two-terminal series parallel (TTSP) graphs.

Let g be a term graph and σ be a list of distinct vertices in g. We call the form $x := [g, \sigma]$ a *binding* for a variable label $x \in X$. A *substitution* θ is a finite collection of bindings $\{x_1 := [g_1, \sigma_1], \ldots, x_n := [g_n, \sigma_n]\}$, where x_i's are mutually distinct variable labels in X and each g_i $(1 \leq i \leq n)$ has no variable labeled with an element in $\{x_1, \ldots, x_n\}$. For a set or a list S, the number of elements in S is denoted by $|S|$. In the same way as logic programming system, we obtain a new term graph f by applying a substitution $\theta = \{x_1 := [g_1, \sigma_1], \cdots, x_n := [g_n, \sigma_n]\}$ to a term graph $g = (V, E, H)$ in the following way. For each binding $x_i := [g_i, \sigma_i] \in \theta$ $(1 \leq i \leq n)$ in parallel, we attach g_i to g by removing the all variables $t_1, \cdots, t_k$ labeled with x_i from H, and by identifying the m-th element of t_j and the m-th element of σ_i for each $1 \leq j \leq k$ and each $1 \leq m \leq |t_j| = |\sigma_i|$, respectively. The resulting term graph f is denoted by $g\theta$. In Figure 2, for example, we draw the term graph $g\theta$ which is obtained by applying a substitution $\theta = \{x := [g_1, (v_1, v_2)], y := [g_2, (w_1, w_2)]\}$ to the term graph g. A *unifier* of two term graphs g_1 and g_2 is a substitution θ such that $g_1\theta$ and $g_2\theta$ are isomorphic. In general, there exists no mgu (most general unifier) of two term graphs. Therefore, in FGS a derivation is based on an enumeration of unifiers and only ground goal is considered in this paper. A graph rewriting rule C is *provable from* an FGS program Γ if C is obtained from Γ by finitely many applications of graph rewriting rules and modus ponens.

3. Refutably Inductive Inference of FGS Programs

In this section, we show that sufficiently large hypothesis spaces of FGS programs are refutably inductive inferable and we present refutably inductive inference algorithms for the hypothesis spaces.

We give our framework of refutably inductive inference of FGS programs according to [1, 5, 7, 9]. Mukouchi and Arikawa [5] originated a computational learning theory of machine discovery from facts and showed that refutable inductive inference is essential in machine discovery from facts and the sufficiently large hypothesis spaces of EFSs are refutably inferable.

Let $g = (V, E, H)$ be a term graph. Then we denote the *size* of g by $|g|$ and define $|g| = |V| + |E| + |H|$. For example, $|g| = |V| + |E| + |H| = 2 + 0 + 2 = 4$ for the term graph $g = (V, E, H)$ in Figure 2. For an atom $p(g_1, \ldots, g_n)$, we define $\|p(g_1, \ldots, g_n)\| = |g_1| + \cdots + |g_n|$. An *erasing binding* is a binding $x := [g, \sigma]$ such that g consists of all vertices in σ, no edge and no variable. An *erasing substitution* is a substitution which contains an erasing binding. In this paper, we disallow an erasing substitution. Then $\|g\theta\| \geq \|g\|$ for any term graph g and any substitution θ.

Definition 1. (1) A graph rewriting rule $A \leftarrow B_1, \ldots, B_m$ is said to be *weakly reducing* if $\|A\theta\| \geq \|B_i\theta\|$ for any $i = 1, \ldots, m$ and any substitution θ. An FGS program Γ is *weakly reducing* if every graph rewriting rule in Γ is weakly reducing.

(2) A graph rewriting rule $A \leftarrow B_1, \ldots, B_m$ is said to be *size-bounded* if $\|A\theta\| \geq \|B_1\theta\| + \cdots + \|B_m\theta\|$ for any substitution θ. An FGS program Γ is *size-bounded* if every graph rewriting rule in Γ is size-bounded.

A size-bounded FGS program is also weakly reducing. For example, the FGS program Γ_{SP} in Figure 1 is weakly reducing but not size-bounded. Let $g = (V, E, H)$ be a term graph. For a variable label $x \in X$, the number of variables in H labeled with x is denoted by $o(x, g)$. For example, $o(x, g) = 1$ and $o(y, g) = 1$ for the term graph $g = (V, E, H)$ in Figure 2. For an atom $p(g_1, \ldots, g_n)$ and a variable label $x \in X$, we define $o(x, p(g_1, \ldots, g_n)) = o(x, g_1) + \cdots + o(x, g_n)$.

A hypothesis space is also called a *class*. We consider the two properties of hypothesis spaces for machine discovery from facts. (Property 1) All hypotheses in a hypothesis space for machine discovery must be recursively enumerable. The following Lemma 1 shows that our target hypothesis spaces have the Property 1. (Property 2) Whether a hypothesis is consistent with examples or not must be recursively decidable. The following Lemma 2 shows that our target hypothesis spaces have the Property 2.

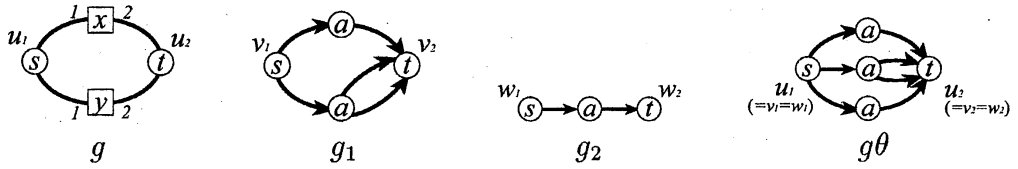

Figure 2: Term graphs g and $g\theta$ obtained by applying a substitution $\theta = \{x := [g_1, (v_1, v_2)], y := [g_2, (w_1, w_2)]\}$ to g.

Lemma 1. (1) A graph rewriting rule $A \leftarrow B_1, \ldots, B_m$ is weakly reducing if and only if $\|A\| \geq \|B_i\|$ and $o(x, A) \geq o(x, B_i)$ for any $i = 1, \ldots, m$ and any variable label x.

(2) A graph rewriting rule $A \leftarrow B_1, \ldots, B_m$ is size-bounded if and only if $\|A\| \geq \|B_1\| + \cdots + \|B_m\|$ and $o(x, A) \geq o(x, B_1) + \cdots + o(x, B_m)$ for any variable label x.

For an FGS program Γ, M_Γ denotes the least Herbrand model of Γ.

Lemma 2. Let Γ be a weakly reducing or size-bounded FGS program. Then the least Herbrand model M_Γ of Γ is a recursively decidable set.

Let Π be a finite set of predicate symbols. We explain the refutably inductive inference of FGS programs. The set of all ground atoms is called the Herbrand base and denoted by HB. For an atom A, $pred(A)$ denotes the predicate symbol of A. For a set $\Pi_0 \subseteq \Pi$ and a set S of atoms, $S\,|_{\Pi_0}$ denotes the set of atoms in S whose predicate symbols are in Π_0. That is $S\,|_{\Pi_0} = \{A \in S \mid pred(A) \in \Pi_0\}$.

A *predicate-restricted complete presentation* of a set $I \subseteq HB$ w.r.t. $\Pi_0 \subseteq \Pi$ is an infinite sequence $(A_1, t_1), (A_2, t_2), \ldots$ of elements in $HB\,|_{\Pi_0} \times \{+, -\}$ such that $\{A_i \mid t_i = +, i \geq 1\} = I\,|_{\Pi_0}$ and $\{A_i \mid t_i = -, i \geq 1\} = HB\,|_{\Pi_0} \setminus I\,|_{\Pi_0}$. A *refutably inductive inference algorithm* (*RIIA*, for short) is a special type of algorithm that receives a predicate-restricted complete presentation as an input. An RIIA $\mathcal{A}$ is said to *refute* a class, if $\mathcal{A}$ produces the sign "refute" as an output and stops. An RIIA either produces infinitely many FGS programs as outputs or refutes a class. For an RIIA $\mathcal{A}$ and a presentation δ, $\mathcal{A}(\delta[m])$ denotes the last output produced by $\mathcal{A}$ which is successively presented the first m elements in δ. An RIIA $\mathcal{A}$ is said to *converge* to an FGS program Γ for a presentation δ, if there is a positive integer m_0 such that for any $m \geq m_0$, $\mathcal{A}(\delta[m])$ is defined and equal to Γ. Let $\mathcal{FC}$ be a class of FGS programs. For an FGS program $\Gamma \in \mathcal{FC}$ and a predicate-restricted complete presentation δ of M_Γ w.r.t. $\Pi_0 \subseteq \Pi$, an RIIA $\mathcal{A}$ is said to be *infer* the FGS program Γ w.r.t. $\mathcal{FC}$ *in the limit from* δ, if $\mathcal{A}$ converges to an FGS program $\Gamma' \in \mathcal{FC}$ with $M_{\Gamma'}\,|_{\Pi_0} = M_\Gamma\,|_{\Pi_0}$ for δ.

Definition 2 ([5]). A class $\mathcal{FC}$ is said to be *theoretical-term-freely and refutably inferable from complete data*, if for any nonempty finite subset Π_0 of Π, there is an RIIA $\mathcal{A}$ which satisfies the following condition: For any set $I \subseteq HB$ and any predicate-restricted complete presentation δ of I w.r.t. Π_0, (i) if there is an FGS program $\Gamma \in \mathcal{FC}$ such that $M_\Gamma\,|_{\Pi_0} = I\,|_{\Pi_0}$, then $\mathcal{A}$ infers Γ w.r.t. $\mathcal{FC}$ in the limit from δ, (ii) otherwise $\mathcal{A}$ refutes the class $\mathcal{FC}$ from δ.

Theoretical terms are supllementary predicates that are necessary for defining some goal predicates. In the above definition, the phrase "theoretical-term-freely inferable" means that from the only facts on the goal predicates an RIIA can generates some suppllementary predicates.

$\mathcal{WR}^{[\leq n]}$ denotes the set of weakly reducing FGS programs at most n graph rewriting rules. $\mathcal{SB}^{[\leq n]}$ denotes the set of size-bounded FGS programs with at most n graph rewriting rules. For constant $n \geq 1$, $\mathcal{WR}^{[\leq n]}$ and $\mathcal{SB}^{[\leq n]}$ have infinitely many hypotheses.

$\mathcal{MWR}^{[m]}[\Pi_0](s)$ denotes the set of all canonical wealky reducing FGS programs with just m graph rewriting rules and predicite symbols in Π_0 such that the head's size of each rule is not greater than s.

By the procedure RIIA_WR in Figure 3, we have (1) of the following Theorem 1 and

```
procedure RIIA_WR(n, Π0);
begin
  T = ∅; F = ∅;
  read_store(T, F);
  while T = ∅ do begin
    output the empty FGS program;
    read_store(T, F);
  end;
  T0 = T; F0 = F;
  for m = 1 to n do begin
    sm = max{||A|| | A ∈ Tm−1};
    recursively generate MWR^[m][Π0](sm),
    and set it to S;
    for each Γ ∈ S do
      while (T, F) is consistent with MΓ
        do begin
          output Γ;
          read_store(T,F);
        end;
    Tm = T; Fm = F;
  end;
  refute the class WR^[≤n] and stop;
end;
procedure read_store(T, F);
begin
  read the next fact (w, t);
  if t =' +' then T = T ∪ {w}
  else F = F ∪ {w};
end.
```

Figure 3: RIIA_WR: a refutably inductive inference algorithm for the class $\mathcal{WR}^{[\leq n]}$

we have (2) by a similar procedure. The proof of Theorem 1 is based on [5].

Theorem 1. (1) For any $n \geq 0$, the class $\mathcal{WR}^{[\leq n]}$ of all weakly reducing FGS programs with at most n graph rewriting rules is theoretical-term-freely and refutably inferable from complete data.

(2) For any $n \geq 0$, the class $\mathcal{SB}^{[\leq n]}$ of all size-bounded FGS programs with at most n graph rewriting rules is theoretical-term-freely and refutably inferable from complete data.

We can construct a machine discovery system for a refutably inferable hypothesis space. Thus the above Theorem 1 gives a theoretical foundation of the KD-FGS system.

4. Implementation of KD-FGS and Experimental Results

We have implemented a prototype of the KD-FGS system by constructing an FGS interpreter and a refutably inductive inference algorithm in Common Lisp. The FGS interpreter is an extension of the Prolog interpreter (P. Norvig[6], Chap. 11).

In Table 1, we summarize 6 experiments of running KD-FGS on a DEC-Alpha compatible workstation (clock 500 MHz) with GCL2.2. In Exp. 1 and 2, input data are positive and negative examples of TTSP graph (see Figure 1). In Exp. 1 (resp., 2), the hypothesis space $\mathcal{C}_1$ (resp., $\mathcal{C}_2$) is the set of all restricted weakly reducing FGS programs with at most 2 (resp., 2) atoms in each body and at most 2 (resp., 3) rules in each program, which is denoted by "#atom≤ 2, #rule≤ 2" (resp., "#atom≤ 2, #rule≤ 3"). After the system receives 3 positive and 5 negative examples, which is denoted by "#pos=3, #neg=5", it refutes $\mathcal{C}_1$ in Exp. 1 (resp., it converges to a correct FGS program in $\mathcal{C}_2$ for TTSP graphs in Exp. 2). We confirm that the system is useful for knowledge discovery from graph data.

5. Refutably Inductive Inference of NLC Graph Grammars

FGS is a kind of graph generating system. So we consider refutable inference of NLC graph grammars.

Definition 3 ([3]).
A node-label controlled graph grammar (NLC graph grammar, for short) is a five-tuple (Σ, N, P, S, C), where

1. Σ is a finite set of *terminal labels*,
2. N is a finite set of *nonterminal labels* with $\Sigma \cap N = \emptyset$,
3. P is a finite set of *productions* each production has the form $X \rightarrow \alpha$, where X is a nonterminal label in N and α is a graph with a vertex label in $\Sigma \cup N$ and an edge label in Λ,
4. S is a special nonterminal label called the *start symbol*,

Table 1: Experimental results on the KD-FGS system

No.	Examples	Hypothesis Space	Received Examples	Result
1	TTSP	#atom≤ 2, #rule≤ 2	#pos=3, #neg=5	refute
2	graph	#atom≤ 2, #rule≤ 3	#pos=3, #neg=5	infer
3	undirected	#atom≤ 1, #rule≤ 2	#pos=3, #neg=6	refute
4	tree	#atom≤ 1, #rule≤ 3	#pos=3, #neg=6	infer
5	directed	#atom≤ 1, #rule≤ 2	#pos=3, #neg=10	refute
6	tree	#atom≤ 1, #rule≤ 3	#pos=4, #neg=10	refute

5. C is a connection relation $C \subseteq (\Sigma \cup N) \times (\Sigma \cup N) \times \Lambda$.

In this paper, we assume that in each production $X \to \alpha$ the graph α has at least one vertex and no multiple edge. Let $G = (\Sigma, N, P, S, C)$ be an NLC graph grammar. The way that a production $X \to \alpha$ in P is applied to transform a graph is as follows:
Step 1. Start with a graph μ and a specific occurrence of an X-node in μ. We call this node the *mother* node. The set of nodes which are directly connected to the mother node is denoted by $Neighborhood$.
Step 2. Delete the mother node and all edges incident to the mother node from the graph μ, and call the resulting graph μ'.
Step 3. Add to μ' a copy of the labeled graph α. This new occurrence of α is called the *daughter* graph.
Step 4. For each pair (Y, Z, L) in the connection relation, connect every Y-node in the daughter graph to every Z-node in $Neighborhood$ by the edge labeled with $L \in \Lambda$.

For example, the NLC graph grammar $G = (\{a\}, \{A\}, \{A \to \alpha, A \to \beta\}, A, \{(A, a, a), (a, a, a)\})$ can generate the set of all complete graphs [2], where α is a graph consisting of one vertex labeled with a and β is a graph consisting of two vertices labeled with A and a, respectively, and one edge labeled with a between them.

Theorem 2. For any $n \geq 0$, the class of all NLC graph grammars with at most n productions is refutably inferable from complete data.

6. Conclusion

We show that sufficiently large hypothesis spaces of FGS programs are refutably inferable and thus give a theoretical foundation of the KD-FGS system, which is a knowledge discovery system from graph data. We also consider refutable inference of NLC graph grammars.

References

[1] S. Arikawa, T. Shinohara, and A. Yamamoto. Learning elementary formal systems. *Theoretical Computer Science*, 95:97–113, 1992.

[2] A. Ehrenfeucht, M. G. Main, and G. Rozenberg. Restrictions on NLC graph grammars. *Theoretical Computer Science*, 31:211–223, 1984.

[3] D. Janssens and G. Rozenberg. On the structure of node-label-controlled graph languages. *Information Sciences*, 20:191–216, 1980.

[4] S. Muggleton and D. Page. Special issue on inductive logic programming. *Machine Learning*, 26(2–3):97–252, 1997.

[5] Y. Mukouchi and S. Arikawa. Towards a mathematical theory of machine discovery from facts. *Theoretical Computer Science*, 137:53–84, 1995.

[6] P. Norvig. *Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp*. Morgan Kaufmann, 1992.

[7] T. Shinohara. Rich class inferable from positive data: length-bounded elementary formal systems. *Information and Computation*, 108:175–186, 1994.

[8] T.Uchida,T.Shoudai,and S.Miyano. Parallel algorithm for refutation tree problem on formal graph systems. *IEICE Transactions on Information and Systems*, E78-D(2):99–112, 1995.

[9] A. Yamamoto. Procedural semantics and negative information of elementary formal system. *Journal of Logic Programming*, 13:89–98, 1992.

[10] K. Yoshida and H. Motoda. Inductive inference by stepwise pair expansion. *Journal of Jpn. Soc. Artif. Intell.*, 12(1):58–67, 1997.