

Inductive Inference from Positive Data is Powerful

Shinohara, Takeshi
Department of Artificial Intelligence Kyushu Institute of Technology

<https://hdl.handle.net/2324/3127>

出版情報 : RIFIS Technical Report. 20, 1989-11-09. Research Institute of Fundamental Information Science, Kyushu University

バージョン :

権利関係 :

RIFIS Technical Report

Inductive Inference from Positive Data is Powerful

Takeshi Shinohara

November 9, 1989

Research Institute of Fundamental Information Science
Kyushu University 33
Fukuoka 812, Japan

Inductive Inference from Positive Data is Powerful

by

Takeshi SHINOHARA
Department of Artificial Intelligence
Kyushu Institute of Technology

Abstract

Inductive inference from positive data is shown to be remarkably powerful using a framework of elementary formal system. An elementary formal system, EFS for short, is a kind of logic program on Σ^+ consisting of finitely many axioms. Any context-sensitive language is definable by a restricted EFS, called length-bounded EFS. Length-bounded EFS's with at most n axioms are considered to show that inductive inference from positive data works successfully for their models as well as for their languages. From this it follows that any class of usual logic programs, like Prolog programs, corresponding to length-bounded EFS's can be inferred from positive facts.

1. Introduction

Inductive inference [3] is a process to guess an unknown rule from its examples. Rules we consider in this paper are subsets of a universal set U . For example, U is the set Σ^+ of all words over an alphabet Σ if rules are languages, or the Herbrand base consisting of all ground atoms if rules are models of logic programs. Positive examples of a rule R are elements in R . Negative examples are the other elements. Inductive inference from positive data is a process to guess a rule when only positive examples are available. We have a well-known theorem by Gold [7] that indicates the weakness of inductive inference from positive data. Immediately from his theorem even the class of all regular languages, which is probably the smallest in Chomsky hierarchy, is not inferable

from positive data. For a long time inductive inference from positive data had received few attentions.

However, in such a situation, Angluin [1, 2] gave a theorem characterizing inferability from positive data and interesting classes. By her theorem we know not a few possibilities remain in inductive inference from positive data, at least in principle. Following her results, several studies have been developed. Nevertheless, the classes shown to be inferable from positive data are too poor and narrow for many people to believe powerfulness of inductive inference from positive data in reality.

Recently the author and his co-workers developed a unifying framework [5] for inductive inference, called *elementary formal system* (EFS for short), which is originally devised by Smullyan [14] to reconstruct recursion theory. In a word, EFS is a kind of logic programming language on Σ^+ [16], and is shown to be natural device to define languages [4]. Using this framework we introduce a hierarchy of language classes that are characterized by a syntactic restriction and the number of axioms. The class of pattern languages, which may be the first interesting class shown to be inferable from positive data [1], is located at the bottom of the hierarchy.

Here we give an example of EFS which is a finite set of definite clauses. Let Γ be $\{ p(a, b, c) \leftarrow, p(ax, by, cz) \leftarrow p(x, y, z), q(xyz) \leftarrow p(x, y, z) \}$, where a, b, c are constant symbols from an alphabet Σ , x, y, z are variables, and p and q are predicate symbols. In EFS's we use two inference rules, one is an application of a substitution for variables by nonempty words, and the other is modus ponens. The language defined by Γ and q is $L(\Gamma, q) = \{ w \in \Sigma^+ \mid p(w) \text{ is provable from } \Gamma \} = \{ a^n b^n c^n \mid n \geq 1 \}$.

A definite clause $A \leftarrow B_1, \dots, B_n$ is called to be *variable-bounded* if variables in B_1, \dots, B_n appear also in A . Variable-bounded EFS is the set of variable-bounded clauses. The restriction on variable-bounded EFS's does not essentially affect to the descriptive power of EFS in the sense that every recursively enumerable language is definable by a variable-bounded EFS. The EFS Γ in the example above is variable-bounded. An definite clause $A \leftarrow B_1, \dots, B_n$ is called to be *length-bounded* if the total length of $B_1\theta, \dots, B_n\theta$ does not exceed the length of $A\theta$ for any substitution θ , where the length of an atom is the sum of the lengths of terms in it. The EFS Γ in the example above is length-bounded. The class of languages definable by length-bounded EFS's coincides with the class of context-sensitive languages.

In this paper we deal with length-bounded EFS's consisting of *at most* n clauses, to show that the class of their minimal models and the class of their

languages are inferable from positive data. Although we put a restriction on the number of clauses in EFS, it is not trivial for such a class to be inferable from positive data because the class consists of infinitely many models or languages. On the other hand, for example, the class of regular languages accepted by finite state automata with at most n states contains only finitely many languages, and therefore its inferability from positive data is obvious.

First we show that the class of such EFS models is inferable from positive data. An observation from our discussion below teaches us that the class of the usual logic programs corresponding to our EFS's is inferable from positive data. However, this result is not enough for us to deal with inference of languages because data in the context of language identification tell nothing about predicates other than one special predicate. For example, no information about predicate symbol p is available from positive examples of $L(\Gamma, q)$ above. Finally we show that our class of EFS languages is inferable from positive data. Interpreting the latter result in terms of usual logic programming, we know that such a class of logic programs can be inferred from examples about one predicate in question without examples for other internal predicates, so called *theoretical terms*.

2. Preliminaries

We start with basic definitions on elementary formal systems according to [5, 16] and brief review of the articles on inductive inference from positive data [1, 2, 12, 13, 15].

2.1 Elementary formal systems

Let Σ , X , and Π be mutually disjoint sets. We assume that Σ and Π are finite. Elements in Σ , X , and Π are called *symbols*, *variables*, and *predicate symbols*, respectively. We denote symbols by a, b, c, \dots , variables by x, y, z, x_1, x_2, \dots , and predicate symbols by p, q, p_1, p_2, \dots . Each predicate symbol is associated with a nonnegative integer called *arity*.

Throughout in this paper we assume Σ , X , and Π are arbitrarily fixed. A^+ denotes the set of all nonempty finite strings over a set A .

DEFINITION A *term* is an element of $(\Sigma \cup X)^+$. Each term is denoted by π, π_1, π_2, \dots . A *ground term* is an element of Σ^+ . Terms are also called *patterns* and ground terms are also called *words*.

DEFINITION An *atomic formula* (or *atom* for short) is an expression of the form $p(\pi_1, \dots, \pi_n)$, where the arity of $p \in \Pi$ is n , and π_1, \dots, π_n are terms. An atom $p(\pi_1, \dots, \pi_n)$ is *ground* if terms π_1, \dots, π_n are all ground.

We denote atoms by $A, A_1, A_2, \dots, B, B_1, B_2, \dots$. *Well-formed formulas, clauses, empty clause* (\square) and *ground clauses* are defined in the ordinal ways [8].

DEFINITION A *definite clause* is a clause of the form

$$A \leftarrow B_1, \dots, B_n,$$

where $n \geq 0$ and A, B_1, \dots , and B_n are atoms. We call atom A the *head* of a clause, and sequence of atoms B_1, \dots, B_n the *body*.

DEFINITION An *elementary formal system* (EFS for short) is a finite set of definite clauses, which are called *axioms*.

DEFINITION A *substitution* is a homomorphism from terms to terms that maps each symbol $a \in \Sigma$ to itself. By $\pi\theta$ we denote the image of a term π by a substitution θ . For an atom $p(\pi_1, \dots, \pi_n)$, and a clause $C = A \leftarrow B_1, \dots, B_n$, we define $p(\pi_1, \dots, \pi_n)\theta = p(\pi_1\theta, \dots, \pi_n\theta)$ and $C\theta = A\theta \leftarrow B_1\theta, \dots, B_n\theta$.

DEFINITION A clause C is *provable from* an EFS Γ , we write $\Gamma \vdash C$, if C is obtained from Γ by finitely many applications of substitutions and modus ponens. That is, we define the relation $\Gamma \vdash C$ inductively as follows:

- (1) If $\Gamma \in C$ then $\Gamma \vdash C$.
- (2) If $\Gamma \vdash C$ then $\Gamma \vdash C\theta$ for any substitution θ .
- (3) If $\Gamma \vdash A \leftarrow B_1, \dots, B_{n+1}$ and $\Gamma \vdash B_{n+1}$ then $\Gamma \vdash A \leftarrow B_1, \dots, B_n$.

DEFINITION For an EFS Γ and $p \in \Pi$ with arity n , we define $L(\Gamma, p) = \{(w_1, \dots, w_n) \in (\Sigma^+)^n \mid \Gamma \vdash p(w_1, \dots, w_n)\}$. If p is unary then $L(\Gamma, p)$ is a language over Σ . A language L is *definable by EFS* or an *EFS language* if such Γ and p exist.

Let $v(A)$ denote the set of all variables in an atom A , $|\pi|$ denote the length of a term π , and $o(x, \pi)$ denote the number of all occurrences of a variable x in a term π . For an atom $p(\pi_1, \dots, \pi_n)$, we define

$$\begin{aligned} |p(\pi_1, \dots, \pi_n)| &= |\pi_1| + \dots + |\pi_n|, \\ o(x, p(\pi_1, \dots, \pi_n)) &= o(x, \pi_1) + \dots + o(x, \pi_n). \end{aligned}$$

DEFINITION A clause $A \leftarrow B_1, \dots, B_n$ is *variable-bounded* if

$$v(A) \supseteq v(B_i) \quad (i=1, \dots, n).$$

An EFS Γ is *variable-bounded* if each axiom of Γ is variable-bounded.

DEFINITION A clause $A \leftarrow B_1, \dots, B_n$ is *length-bounded* if

$$|A\theta| \geq |B_1\theta| + \dots + |B_n\theta|$$

for any substitution θ . An EFS Γ is *length-bounded* if each axiom of Γ is length-bounded.

The concept of length-boundedness is characterized by the following Lemma, by which we know that length-bounded clauses are all variable-bounded.

LEMMA 1 [5] A clause $A \leftarrow B_1, \dots, B_n$ is length-bounded if and only if

$$|A| \geq |B_1| + \dots + |B_n|, \text{ and} \\ o(x, A) \geq o(x, B_1) + \dots + o(x, B_n)$$

for any variable x .

Here we should note that each substitution may not erase any variable, that is, $x\theta$ may not be empty word for any variable x . This is an essential point for our discussion in this paper. We need another discussion when we allow erasing substitutions as in [].

LEMMA 2 If a clause $C = A \leftarrow B_1, \dots, B_n$ is provable from a length-bounded EFS and Γ , then C is length-bounded.

The following theorem says that variable-bounded EFS's are powerful enough to define languages.

THEOREM 3 [5] A language $L \subseteq \Sigma^+$ is definable by a variable-bounded EFS if and only if L is recursively enumerable.

The class of languages defined by length-bounded EFS's is characterized by the following theorem.

THEOREM 4 [5] A language $L \subseteq \Sigma^+$ is definable by a length-bounded EFS if and only if L is context-sensitive.

The *Herbrand base*, we denote by HB , is the set of all ground atoms. Every subset $I \subseteq HB$ is called an *Herbrand interpretation*. The *minimal model* of an EFS Γ is given by

$$M(\Gamma) = \{ M \subseteq HB \mid M \text{ is an Herbrand interpretation of } \Gamma \}.$$

EFS model is an Herbrand interpretation I such that $I = M(\Gamma)$ for some EFS Γ . As for usual logic programming languages, we can use a derivation procedure based on resolution principle. We define the *success set* of an EFS Γ by

$$SS(\Gamma) = \{ A \in HB \mid \text{there exists a refutation from } \leftarrow A \},$$

and the *provable set* of Γ by

$$PS(\Gamma) = \{ A \in HB \mid \Gamma \vdash A \}.$$

On the relation between three set defined above, the following theorem is known.

THEOREM 5 [5, 16] $M(\Gamma) = SS(\Gamma) = PS(\Gamma)$.

By this theorem we need not distinguish $M(\Gamma)$, $SS(\Gamma)$, and $PS(\Gamma)$ from each other. Further more about length-bounded EFS's, we have the following theorem, which is based on the fact that we can set a time bound for the derivation procedure.

THEOREM 6 [5, 16] If an EFS Γ is length-bounded then $M(\Gamma)$ is recursive.

2.2 Inductive inference from positive data

We give basic definitions and results on inductive inference from positive data in slightly modified forms, since we will discuss in this paper inductive inference of EFS languages as well as EFS models.

Let U be a set to which we refer as a *universal set*. A subset $R \subseteq U$ is called a *rule*. When we are dealing with EFS languages the universal set U is the set Σ^+ . For EFS models U is the Herbrand base HB .

DEFINITION A class of rules $C = R_1, R_2, \dots$ is said to be an *indexed family of recursive rules* if there exists a computable function $f: N \times U \rightarrow \{0, 1\}$ such that

$$f(i, s) = \begin{cases} 1, & \text{if } s \in R_i \\ 0, & \text{otherwise.} \end{cases}$$

When rules are EFS languages, the index i of R_i can be considered as a pair of an EFS Γ and a predicate symbol p with arity 1 such that $L(\Gamma, p) = R_i$. As the index of an EFS model we can use an EFS Γ such that $M(\Gamma) = R_i$. From here on, the classes of rules are assumed to be an indexed family of recursive rules.

DEFINITION A *complete presentation* of a rule R is an infinite sequence $(s_1, t_1), (s_2, t_2), \dots$ such that t_i is 0 or 1, $\{s_i \mid t_i = 1\} = R$, and $\{s_i \mid t_i = 0\} = U - R$. A *positive presentation* of a nonempty rule R is an infinite sequence of s_1, s_2, \dots such that $\{s \mid s = s_i \text{ for some } i\} = R$.

An *inference machine* is an effective procedure that requests input from time to time and produces output from time to time. An output produced by an inference machine is called a *guess*. Let $\sigma = s_1, s_2, \dots$ be an infinite sequence, and g_1, g_2, \dots be the sequence of guesses produced by an inference machine IM when

elements of σ are successively given to IM . Then we say that IM on input σ converges to g , if the sequence g_1, g_2, \dots of guesses is finite and ends with g , or there exists a positive integer k_0 such that $g_k = g$ for all $k \geq k_0$.

DEFINITION A class of rules $C = R_1, R_2, \dots$ is said to be *inferable from positive (or complete) data* if there exists an inference machine IM such that IM on input σ converges to g with $R_g = R_i$ for any index i and any positive (or complete) presentation σ of R_i .

Gold [7] showed that any indexed family of recursive rules is inferable from complete data. He also proved that inference from positive data is not possible for any class of rules that contains all finite rules and at least one infinite rule. By his theorem we can easily show that even the class of regular languages is not inferable from positive data. By this result most researchers in the field of grammatical inference had been disappointed until Angluin [1, 2] gave a new life to inductive inference from positive data by presenting a theorem, which characterizes classes inferable from positive data, and nontrivial such classes including pattern languages.

Here we give one of the sufficient conditions shown by her for inferability from positive data. Using this condition she proved that the class of pattern languages is inferable from positive data. For more details the reader should be referred to literatures [1, 2]. We denote the number of elements in a set S by $\#S$.

DEFINITION A class C has *finite thickness* if $\#\{R \in C \mid A \in R\}$ is finite for any $A \in U$.

THEOREM 7 [1, 2] If a class C has finite thickness then C is inferable from positive data.

The author showed in his previous work [12] that unions of two pattern languages are inferable from positive data. Wright [15] extended this result to unions of three or more languages by showing that the following condition is sufficient for inferability from positive data and it is closed under union of languages.

DEFINITION. A class C has *infinite elasticity* if there exist two infinite sequences A_0, A_1, A_2, \dots and R_1, R_2, R_3, \dots , where $A_i \in U, R_i \in C$, such that $A_j \in R_k$ if and only if $j < k$. C has *finite elasticity* if C does not have infinite elasticity.

Here we should note that A_0, A_1, A_2, \dots and R_1, R_2, R_3, \dots in the definition above have to be mutually distinct.

LEMMA 8 [15] If a class C has finite thickness then C has finite elasticity.

THEOREM 9 [15] If both of classes C_1 and C_2 have finite elasticity then the class of unions $C = \{R_1 \cup R_2 \mid R_1 \in C_1 \text{ and } R_2 \in C_2\}$ has finite elasticity.

THEOREM 10 [15] If a class C has finite elasticity then C is inferable from positive data.

The author also showed a theorem, which is one of the special cases of our main result in this paper, that the class of languages definable by EFS with at most two axioms is inferable from positive data [13]. Such a result on EFS's does not follow immediately from Theorem 9, because models and languages of EFS's are quite different from simple unions.

3. Inductive inference of EFS models from positive data

DEFINITION An EFS Γ is *reduced with respect to* an Herbrand interpretation I if $I \subseteq M(\Gamma)$ but $I \not\subseteq M(\Gamma')$ for any $\Gamma' \subsetneq \Gamma$.

LEMMA 11 Let Γ be a length-bounded EFS and $C = A \leftarrow B_1, \dots, B_n$ be a clause provable from Γ . Then the head of every axiom used to prove C is not longer than the head A of C .

Proof We show by mathematical induction on the number of applications of inference rule. Note that all clauses including C provable from Γ are length-bounded by Lemma 2.

If we can prove a clause C by only one application of inference rule, then the clause C itself is an axiom of Γ .

Otherwise, the rule lastly used to prove C should be an application of a substitution or modus ponens. If $C = C' \theta$ for some provable clause C' and some substitution θ , then the head of C' is not longer than the head of C . If $C' = A \leftarrow B_1, \dots, B_n, B_{n+1}$ and B_{n+1} are provable, then the head of C' and B_{n+1} are not longer than the head of C . Therefore, it is clear that any head of axioms used to prove C' and B_{n+1} is not longer than the head of C . \square

LEMMA 12 Let $I = \{A_1, \dots, A_k\}$ be a nonempty finite Herbrand interpretation and Γ be a length-bounded EFS that is reduced with respect to I . Then for any axiom $A \leftarrow B_1, \dots, B_n$ of Γ , $|A| \leq \max\{|A_1|, \dots, |A_k|\}$.

Proof If an EFS Γ contains an axiom $A \leftarrow B_1, \dots, B_n$ such that $|A| > \max\{|A_1|, \dots, |A_k|\}$. Then, clearly from Lemma 11, the axiom can never be used to prove A_i

from Γ . This contradicts our assumption that Γ is reduced with respect to $\{A_1, \dots, A_k\}$. \square

From Lemma 12 and the fact that the number of patterns shorter than a fixed length is finite, we have the following, where we ignore duplicated occurrences of atoms in the body of a clause.

LEMMA 13 Given a nonempty finite Herbrand interpretation I , there exist only finitely many length-bounded EFS's that are reduced with respect to I .

THEOREM 14 For any $n \geq 1$, the class $M^n = \{M(\Gamma) \mid \Gamma \text{ is length-bounded EFS, } \#\Gamma \leq n, M(\Gamma) \neq \emptyset\}$ is inferable from positive data.

Proof Note that by Theorem 6 the class M^n is an indexed family of recursive rules. By mathematical induction on n , we show M^n has finite elasticity.

If $A \in M(\Gamma)$, $\#\Gamma=1$, then $\Gamma = \{B\}$, B is an atom with the same predicate symbol as A , and $A = B\theta$ for some substitution. Therefore $|B| \leq |A|$. The number of such atoms is finite except renaming of variables. Thus M^1 has finite thickness, and therefore by Lemma 8, finite elasticity.

For any $n < i$ ($i \geq 2$), we assume that M^n has finite elasticity. Let M^i have infinite elasticity. Then there exist an infinite sequence of ground atoms A_0, A_1, A_2, \dots and an infinite sequence of length-bounded EFS's $\Gamma_1, \Gamma_2, \Gamma_3, \dots$ such that $\#\Gamma_k \leq i$ and $A_j \in M(\Gamma_k)$ if and only if $j < k$. Let h be a function defined by

$$h(k) = \min \{j \leq k \mid \Gamma_k \text{ is reduced with respect to } \{A_0, \dots, A_j\} \text{ or } j = k\}.$$

We consider two cases depending on whether the set $\{h(k) \mid k = 1, 2, \dots\}$ has a finite bound or not.

Case 1. If $\{h(k) \mid k = 1, 2, \dots\}$ has a finite bound j_0 such that $h(k) \leq j_0$ for all k . Then Γ_k should be reduced with respect to $\{A_0, \dots, A_{j_0}\}$ for all $k > j_0$. However, Lemma 13 claims that the number of such EFS's is finite. This contradicts our assumption on the infinite elasticity of M^i .

Case 2. If $\{h(k) \mid k = 1, 2, \dots\}$ does not have any finite bound, then it contains an infinitely ascending sequence $1 < h(k_1) < h(k_2) < \dots$ such that $k_1 < k_2 < \dots$. Clearly Γ_{k_j} is not reduced with respect to $\{A_0, \dots, A_{h(k_j)-1}\}$. Therefore there exists $\Gamma'_{k_j} \subsetneq \Gamma_{k_j}$ such that $\{A_0, \dots, A_{h(k_j)-1}\} \subseteq M(\Gamma'_{k_j})$ but $A_{h(k_j)} \notin M(\Gamma'_{k_j})$. Here we should note $\#\Gamma'_{k_j} \leq i-1$. Thus we have two infinite sequences

$$A_0, A_{h(k_1)}, A_{h(k_2)}, \dots \text{ and } \Gamma'_{k_1}, \Gamma'_{k_2}, \Gamma'_{k_3}, \dots$$

that show the infinite elasticity of M^{i-1} . This is a contradiction to the inductive hypothesis.

Since we can show a contradiction in each case, M^n has finite elasticity for any $n \geq 1$. Therefore by Theorem 10 M^n is inferable from positive data. \square

3. Inductive inference of EFS languages from positive data

We have just proved in the previous section that length-bounded EFS models with at most n axioms are inferable from positive data. However this result is not sufficient for us to deal with languages. To infer an EFS model, we can use examples about all predicate symbols in it, as in MIS by Shapiro [9]. On the other hand, identification of an EFS languages should be worked on words, which can tell nothing about predicate symbols but a special one. Fortunately we have the following theorem that guarantees us to successfully infer languages definable by length-bounded EFS's with at most n axioms from positive data.

THEOREM 15 For any $n \geq 1$, the class $L^n = \{ L(\Gamma, p) \mid \Gamma \text{ is length-bounded, } p \text{ is unary, } \#\Gamma \leq n, L(\Gamma, p) \neq \emptyset \}$ is inferable from positive data.

Proof We should note that by Theorem 6 the class L^n is also an indexed family of recursive rules. Assume L^n has infinite elasticity. Then there exist an infinite sequence of words w_0, w_1, w_2, \dots and an infinite sequence of pairs of length-bounded EFS and predicate symbol $(\Gamma_1, p), (\Gamma_2, p), (\Gamma_3, p), \dots$ such that $\#\Gamma_k \leq i$ and $w_j \in L(\Gamma_k, p)$ if and only if $j < k$. Here it should be noted that we can assume that all languages in the infinite sequence are defined by a common predicate symbol p without any loss of generality. It is clear that infinite sequence of ground atoms $p(w_0), p(w_1), p(w_2), \dots$ and infinite sequence of length-bounded EFS's $\Gamma_1, \Gamma_2, \Gamma_3, \dots$ show the infinite elasticity of M^n . Thus we have a contradiction. Therefore L^n has finite elasticity, and again by Theorem 10, L^n is inferable from positive data. \square

4. Discussion

Thus we can reveal the powerfulness of inductive inference from positive data. Angluin and Smith [3] pointed out that Chomsky hierarchy might not be suited for inductive inference. In our framework of EFS's we can construct a new hierarchy remarkably suitable for inductive inference as well as Chomsky hierarchy.

In the usual logic programming languages such as Prolog, almost the same discussion is valid for any class of logic programs corresponding to our EFS's. Such a program is called *linear* in [10] or *reducing* in [6]. Thus we can translate Theorem 14 into another theorem in terms of logic programming.

Acknowledgments

The author wishes to thank Prof. Setsuo Arikawa for presenting him a good framework, elementary formal system, to study language learning. He also would like to appreciate fruitful discussions on logic programs with Akihiro Yamamoto and Hiroki Arimura.

References

- [1] Angluin, D.: Finding common patterns to a set of strings, Proc. 11th Annual ACM Symp. Theory of Computing, 130 - 141, 1979.
- [2] Angluin, D.: Inductive inference of formal languages from positive data, Inform. Contr., 45, 117 - 135, 1980.
- [3] Angluin, D. and Smith, C. H.: Inductive inference: Theory and methods, Computing Surveys, 15, 237 - 269, 1983.
- [4] Arikawa, S.: Elementary formal systems and formal languages - simple formal systems, Memoirs of Fac. Sci., Kyushu Univ. Ser. A, Math., 24, 47 - 75, 1970.
- [5] Arikawa, S., Shinohara, T. and Yamamoto, A.: Elementary formal system as a unifying framework for language learning, Proc. 2nd Workshop Comput. Learning Theory, 312 - 327, 1989.
- [6] Arimura, H.: Completeness of depth-bounded resolution in logic programming, Proc. 6th Conf, Japan Soc. Software Sci. Tech., 61 - 64, 1989.
- [7] Gold, E.M.: Language Identification in the Limit, Inf. & Contr., 10, 447 - 474, 1967.
- [8] Lloyd, J.W.: *Foundations of Logic Programming*, Springer - Verlag, 1984.
- [9] Shapiro, E.Y.: Inductive Inference of Theories From Facts, YALEU / DCS / TR -192, 1981.
- [10] Shapiro, E.Y.: Alternation and the computational complexity of logic programs, J. Logic Program., 1, 19 - 33, 1984.
- [11] Shinohara, T.: Polynomial time inference of extended regular pattern languages, LNCS, 147, 115 - 127, 1983.
- [12] Shinohara, T.: Inferring unions of two pattern languages, Bull. Inf. Cybern., 20, 83 - 88, 1983.
- [13] Shinohara, T.: Inductive inference of formal systems from positive data, Bull. Inf. Cybern., 22, 9 - 18, 1986.
- [14] Smullyan, R. M.: *Theory of Formal Systems*, Princeton Univ. Press, 1961.
- [15] Wright, K.: Identification of unions of languages drawn from an identifiable class, Proc. 2nd Workshop Comput. Learning Theory, 328 - 333, 1989.
- [16] Yamamoto, A.: Elementary formal system as a logic programming language, Proc. Logic Program. Conf. '89, ICOT, 123 - 132, 1989.