Inductive Inference of Monotonic Formal Systems from Positive Data

Shinohara, Takeshi
Department of Artificial Intelligence Kyushu Institute of Technology

出版情報：RIFIS Technical Report. 29, 1990-04-26. Research Institute of Fundamental Information Science, Kyushu University
バージョン：
権利関係：

RIFIS-TR-CS-29

RIFIS Technical Report

Inductive Inference of Monotonic Formal Systems from Positive Data

Takeshi Shinohara

April 26, 1990

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

E-mail: shino@ai.kyutech.ac.jp Phone: 0948 (29)7621

INDUCTIVE INFERENCE OF MONOTONIC FORMAL SYSTEMS FROM POSITIVE DATA

Takeshi Shinohara
Department of Artificial Intelligence
Kyushu Institute of Technology
Iizuka 820, JAPAN
shino@ai.kyutech.ac.jp

Abstract

A formal system is a finite set of expressions, such as a grammar or a Prolog program. A semantic mapping from formal systems to concepts is said to be monotonic if it maps larger formal systems to larger concepts. A formal system Γ is said to be reduced with respect to a finite set X if the concept defined by Γ contains X but the concepts defined by any proper subset Γ' of Γ cannot contain some part of X. Assume a semantic mapping is monotonic and formal systems consisting of at most n expressions that are reduced with respect to X can define only finitely many concepts for any finite set X and any n. Then, the class of concepts defined by formal systems consisting of at most n expressions is shown to be inferable from positive data. As corollaries, the class of languages defined by length-bounded elementary formal systems consisting of at most n axioms, the class of languages generated by context-sensitive grammars consisting of at most n productions, and the class of minimal models of linear Prolog programs consisting of at most n definite clauses are all shown to be inferable from positive data.

1 Introduction

Inductive inference [3] is a process to guess an unknown concept from its examples. What we call concepts here are subsets of a universe U of objects. For example, if we take the set Σ^+ of all words over an alphabet Σ as the universe U, then concepts are so called formal languages. Positive examples of a concept R are elements in R. Negative examples are the others. Inductive inference from positive data is a process to guess a concept when only positive examples are available. We have a well-known theorem by Gold [7] that indicates weakness of inductive inference from positive data. Immediately from this theorem we know even the class of all regular languages, which

is probably the smallest in Chomsky hierarchy, cannot be inferred from positive data. For more than ten years after Gold showed this, inductive inference from positive data had received few attentions.

Such a situation was broken down by Angluin [1, 2] who gave a theorem characterizing inferability from positive data and presented nontrivial classes. The class of pattern languages [1] is one of the most interesting classes shown by her to be inferable. From her results we know not a few possibilities remain in inductive inference from positive data, at least in principle. Following her, several studies have been developed. Nevertheless discovered classes are too poor and narrow for many people to believe powerfulness of inductive inference from positive data in reality.

The author and his co-workers developed a unifying framework [5] for language learning, called *elementary formal system* (*EFS* for short), which is introduced by Smullyan [16] to reconstruct recursion theory. In a word, EFS is a logic program over Σ^+ [18]. Arikawa showed EFS's can be used as a natural device to define formal languages [4]. Using this framework we can introduce a hierarchy of language classes that are characterized by a syntactic restriction and the number of clauses. The class of pattern languages [1] is located at the bottom of the hierarchy.

An EFS is a finite set of definite clauses. For example,

$$\Gamma = \{ p(a, b, c) \leftarrow ;\ p(ax, by, cz) \leftarrow p(x, y, z);\ q(xyz) \leftarrow p(x, y, z) \}$$

is an EFS, where a, b, c are constant symbols taken from an alphabet Σ, x, y, z are variables, and p, q are predicate symbols. Finite strings consisting of constant symbols and variables are also called patterns. 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 $L(\Gamma, q)$ defined by Γ and q is

$$\{ w \in \Sigma^+ \mid q(w) \text{ is provable from } \Gamma \} = \{ a^n b^n c^n \mid n \geqq 1 \}.$$

A definite clause $A \leftarrow B_1, \ldots, B_n$ is called *length-bounded* if the total length of $B_1\theta$, $\ldots$, $B_n\theta$ 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 patterns 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.

Recently the author revealed powerfulness of inductive inference from positive data [15], by showing that the class of languages definable by length-bounded EFS's consisting of at most n clauses is inferable from positive data for any n. Here we should note that the restriction on the number of clauses in EFS does not trivially imply inferability from positive data because the class contains infinitely many languages. On the contrast, 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 is obvious.

This paper is devoted to extend the previous results [15]. First we show a sufficient condition for inferability from positive data in a more general setting. A *concept defining framework* is specified by a triple (U, E, M) of a universe U of objects, a universe E of expressions, and a semantic mapping M from finite sets of expressions to concepts. A finite set of expressions is called a *formal system*. We say M is *monotonic* if for any formal systems Γ' and Γ, $\Gamma' \subseteq \Gamma$ implies $M(\Gamma') \subseteq M(\Gamma)$. A formal system $\Gamma \subseteq E$ is said to be *reduced with respect to* a finite set $X \subseteq U$, if $X \subseteq M(\Gamma)$ but $X \not\subseteq M(\Gamma')$ for any $\Gamma' \subsetneq \Gamma$. We say a concept defining framework (U, E, M) has *bounded finite thickness*, if M is monotonic, and for any finite set $X \subseteq U$ and any $n \geqq 0$

$$\{R \subseteq U \mid R = M(\Gamma), \Gamma \subseteq E, \#\Gamma \leqq n, \Gamma \text{ is reduced with respect to } X\}$$

consists of finitely many concepts. In any concept defining framework (U, E, M) that has bounded finite thickness, the class of concepts defined by formal systems consisting of at most n expressions is shown to be inferable from positive data for any n.

Then, we apply this general result to several concept defining frameworks. As corollaries, not only the class of languages definable by length-bounded EFS's with at most n clauses, but also the class of context-sensitive languages generated by grammars consisting of at most n productions, and the class of minimal models of linear Prolog programs [11] or reducing Prolog programs [6] consisting of at most n definite clauses are all shown to be inferable from positive data.

2 Preliminaries

We start with a brief review of basic definitions and results on inductive inference according to [1, 2, 7, 17] in slightly modified forms to be appropriate for our discussions.

Let U and E be sets, whose elements are called *objects* and *expressions*, respectively. A *concept* is a subset $R \subseteq U$. A *formal system* is a finite subset $\Gamma \subseteq E$. A *semantic mapping* is a mapping M from formal systems to concepts. When $M(\Gamma) = R$, we say a formal system Γ *defines* a concept R or R is a *semantics* of Γ.

DEFINITION A *concept defining framework* is a triple (U, E, M) of a universe U of objects, a universe E of expressions, and a semantic mapping M.

To deal with context-sensitive languages as concepts, for example, we may use nonempty finite strings over an alphabet Σ as objects and context-sensitive productions

as expressions. Here after in this section we fix a concept defining framework (U, E, M) arbitrarily.

DEFINITION A class of concepts $C = R_1, R_2, \ldots$ is said to be an *indexed family of recursive concepts* 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 concepts in a class C are defined by formal systems, the index i of R_i can be considered as a formal system Γ such that $M(\Gamma) = R_i$. From here on, we assume that classes of concepts are an indexed family of recursive concepts.

DEFINITION A *complete presentation* of a concept R is an infinite sequence (s_1,t_1), (s_2,t_2), ... 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 concept R is an infinite sequence of s_1, s_2, ... 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 $\sigma=s_1, s_2, \ldots$ be an infinite sequence, and $g_1, g_2, \ldots$ 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 $g_1, g_2, \ldots$ of guesses is finite and ends with g, or there exists a positive integer k_0 such that $g_k = g$ for all $k \geqq k_0$.

DEFINITION A class of concepts $C = R_1, R_2, \ldots$ 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 concepts is inferable from complete data. He also proved that inference from positive data is impossible for any class of concepts that contains all finite concepts and at least one infinite concept. 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 proving a theorem, which characterizes classes inferable from positive data, and presenting nontrivial classes including the class of pattern languages.

Here we give one of the sufficient conditions for classes to be inferable from positive data shown by her. Using this condition the class of pattern languages is proved to be

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 object $A \in U$.

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

The author showed in his previous work [13] that unions of two pattern languages are inferable from positive data. Wright [17] 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 unions.

DEFINITION A class C has *infinite elasticity* if there exist two infinite sequences A_0, A_1, ... and $R_1, R_2, \ldots$, where $A_i \in U, R_i \in C$, such that for any $k \geqq 1$

$$\{A_0, A_1, \ldots, A_{k-1}\} \subseteq R_k \text{ but } A_k \notin R_k.$$

C has *finite elasticity* if C does not have infinite elasticity.

Here we should note that A_0, A_1, A_2, ... and R_1, R_2, R_3, ... in the definition above have to be pairwise inequivalent.

LEMMA 2 [17] If a class C has finite thickness then C has finite elasticity.

THEOREM 3 [17] 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 4 [17] If a class C has finite elasticity then C is inferable from positive data.

REMARK The condition "$\{A_0, A_1, \ldots, A_{k-1}\} \subseteq R_k$ but $A_k \notin R_k$" in the definition of infinite elasticity is stated as

$$A_j \in R_k \text{ if and only if } j < k,$$

in [17]. Note that our definition does not care whether A_j is contained in R_k or not for any $j > k$. However, if we adopt the original definition, it is shown that finite elasticity is not sufficient for inferability from positive data [9].

In [14] the author showed a theorem, which is one of the special cases of our results in this paper, that the class of languages defined by elementary formal systems consisting of two axioms is inferable from positive data. Such a result on formal

systems does not follow immediately from Theorem 3, because formal systems may define concepts that cannot be represented by simple unions.

3 Inductive Inference of Monotonic Formal Systems

DEFINITION A semantic mapping M is *monotonic* if

$$\Gamma' \subseteq \Gamma \text{ implies } M(\Gamma') \subseteq M(\Gamma).$$

DEFINITION A formal system Γ is *reduced with respect to* a set $X \subseteq U$ if

$$X \subseteq M(\Gamma) \text{ but } X \not\subseteq M(\Gamma') \text{ for any } \Gamma' \subsetneq \Gamma.$$

Intuitively, when a formal system Γ does not have any redundant expressions to cover all objects in X, Γ is said to be reduced with respect to X.

DEFINITION A concept defining framework (U, E, M) has *bounded finite thickness* if M is monotonic , and

$$\#\{ M(\Gamma) \mid \Gamma \text{ is reduced with respect to } X, \#\Gamma \leqq n \} < \infty$$

for any finite set $X \subseteq U$ and any $n \geqq 0$.

Here we should note that bounded finite thickness is a natural extension of finite thickness. That is, if the class of all concepts in a concept defining framework (U, E, M) has finite thickness then (U, E, M) has bounded finite thickness. However the converse does not hold in general. The following is the main result of this paper.

THEOREM 5 Let a concept defining framework (U, E, M) have bounded finite thickness and $C^n = \{ M(\Gamma) \mid \Gamma \subseteq E, \ \#\Gamma \leqq n \}$. Then, the class C^n is inferable from positive data for any $n \geqq 0$.

Proof By mathematical induction on n, we show C^n has finite elasticity.

For the base case $n = 0$, finite elasticity is trivial. Because C^0 contains only one concept.

For any $n < i$ ($i \geqq 1$), we assume that C^n has finite elasticity. Let C^i have infinite elasticity. Then there exist an infinite sequence of objects $A_0, A_1, \ldots$ and an infinite sequence of formal systems $\Gamma_1, \Gamma_2, \ldots$ such that for any $k \geqq 1$, $\#\Gamma_k \leqq i$, $\{ A_0, A_1, \ldots, A_{k-1} \} \subseteq M(\Gamma_k)$ but $A_k \notin M(\Gamma_k)$. Let h be a function defined by

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

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

Case 1 If $\{ h(k) \mid k = 1, 2, \ldots \}$ has a finite bound j_0 such that $h(k) \leqq j_0$ for all k. Then for any $k > j_0$, Γ_k should be reduced with respect to $\{ A_0, \ldots, A_{j_0} \}$. However, bounded finite thickness of (U, E, M) claims that the number of concepts defined by such formal systems is finite. This contradicts our assumption on the infinite elasticity of C^i.

Case 2 If $\{ h(k) \mid k = 1, 2, \ldots \}$ does not have any finite bound, then it contains an infinitely ascending sequence $1 < h(k_1) < h(k_2) < \ldots$ such that $k_1 < k_2 < \ldots$. Let $X = \{A_0, \ldots, A_{h(k_j)}\}$ and $X' = \{A_0, \ldots, A_{h(k_j)-1}\}$.

If Γ_{k_j} is reduced with respect to X, then $X \not\subseteq M(\Gamma'_{k_j})$ for any $\Gamma'_{k_j} \subsetneq \Gamma_{k_j}$. However $X' \subseteq M(\Gamma'_{k_j})$ for some $\Gamma'_{k_j} \subsetneq \Gamma_{k_j}$ because Γ_{k_j} is not reduced with respect to X'. Therefore there exists $\Gamma'_{k_j} \subsetneq \Gamma_{k_j}$ such that $X' \subseteq M(\Gamma'_{k_j})$ but $A_{h(k_j)} \notin M(\Gamma'_{k_j})$.

Otherwise, if Γ_{k_j} is not reduced with respect to X, then $h(k_j) = k_j$. From our assumption on infinite elasticity of C^i, $A_{h(k_j)} = A_{k_j} \notin M(\Gamma_{k_j})$. Since Γ_{k_j} is not reduced with respect to X', $X' \subseteq M(\Gamma'_{k_j})$ for some $\Gamma'_{k_j} \subsetneq \Gamma_{k_j}$.

Therefore there always exists a formal system Γ'_{k_j} such that $\Gamma'_{k_j} \subsetneq \Gamma_{k_j}$ and $X' \subseteq M(\Gamma'_{k_j})$ but $A_{h(k_j)} \notin M(\Gamma'_{k_j})$. Here we should note $\#\Gamma'_{k_j} \leqq i-1$. Thus we have two infinite sequences

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

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

Since we can show a contradiction in each case, C^n has finite elasticity for any $n \geqq 0$. Therefore by Theorem 4 C^n is inferable from positive data. □

4 Corollaries

In the previous section we have shown a sufficient condition for inferability from positive data in an abstract way. In this section we apply it to several classes of concepts. First the class of languages defined by elementary formal systems is considered. General results in the previous section are extracted from those for EFS languages [15]. Then, the class of context-sensitive languages and the class of minimal models of Prolog programs called linear in [11] or reducing in [6] are considered.

4.1 Elementary Formal Systems

Let Σ, X, and Π be mutually disjoint sets. We assume that Σ is finite. Elements in Σ, X, and Π are called *symbols*, *variables*, and *predicate symbols*, respectively. Each predicate symbol is associated with a nonnegative integer called *arity*. We assume a special predicate symbol p_0 with arity 1. A^+ denotes the set of all nonempty finite strings over a set A.

DEFINITION A *term* is an element of $(\Sigma \cup X)^+$. 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 $p(\pi_1, \ldots, \pi_n)$, where the arity of $p \in \Pi$ is n, and $\pi_1, \ldots, \pi_n$ are terms. An atom $p(\pi_1, \ldots, \pi_n)$ is *ground* if terms $\pi_1, \ldots, \pi_n$ are all ground.

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

$$A \leftarrow B_1, \ldots, B_n,$$

where $n \geqq 0$ and $A, B_1, \ldots$, and B_n are atoms.

DEFINITION An *elementary formal system* (*EFS* for short) is a finite set of definite clauses.

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 $A = p(\pi_1, \ldots, \pi_n)$ and a clause $C = A \leftarrow B_1, \ldots, B_n$, we define $A\theta = p(\pi_1\theta, \ldots, \pi_n\theta)$ and $C\theta = A\theta \leftarrow B_1\theta, \ldots, B_n\theta$. A *renaming of variables* is a substitution θ such that $x\theta$ is a variable for any variable x and $x \neq y$ implies $x\theta \neq y\theta$ for any variables x and y.

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

DEFINITION For an EFS Γ and $p \in \Pi$ with arity n, we define $L(\Gamma, p) = \{ (w_1, \ldots, w_n) \in (\Sigma^+)^n \mid \Gamma \vdash p(w_1, \ldots, w_n) \leftarrow \}$. 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. $L(\Gamma, p_0)$ is abbreviated to $L(\Gamma)$.

From definitions it is clear that the semantic mapping L for EFS's is monotonic.

DEFINITION A *renaming of predicate* is a one-to-one mapping $h : \Pi \rightarrow \Pi$ preserving arity. For an atom $p(\pi_1, \ldots, \pi_n)$ and a clause $C = A \leftarrow B_1, \ldots, B_n$, we define $h(p(\pi_1, \ldots, \pi_n)) = h(p)(\pi_1, \ldots, \pi_n)$ and $h(C) = h(A) \leftarrow h(B_1), \ldots, h(B_n)$.

DEFINITION Let Γ_1 and Γ_2 be EFS's. If $\Gamma_1 = \{ h(C)\theta \mid C \in \Gamma_2 \}$ for some renaming of variables θ and some renaming of predicates h such that $h(p_0) = p_0$, then we say Γ_1 is *equivalent to* Γ_2, and denote it by $\Gamma_1 \equiv \Gamma_2$.

LEMMA 6 If $\Gamma_1 \equiv \Gamma_2$, then $L(\Gamma_1) = L(\Gamma_2)$.

Let $|\pi|$ denote the length of a term π. For an atom $p(\pi_1, \ldots, \pi_n)$, we define

$$|p(\pi_1, \ldots, \pi_n)| = |\pi_1| + \cdots + |\pi_n|.$$

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

$$|A\theta| \geqq |B_1\theta| + \cdots + |B_n\theta|$$

for any substitution θ. An EFS Γ is *length-bounded* if axioms of Γ are all length-bounded.

Here we should note that any 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 here. We need another discussion when we allow erasing substitutions as in [12].

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

THEOREM 7 [5] A language $L \subseteq \Sigma^+$ is definable by a length-bounded EFS if and only if L is context-sensitive.

THEOREM 8 [5, 18] Let $\Gamma_1, \Gamma_2 \ldots$ be any recursive enumeration of length-bounded EFS's. Then the class $C = L(\Gamma_1), L(\Gamma_2), \ldots$ is an indexed family of recursive languages.

LEMMA 9 Let $X \subseteq \Sigma^+$ be finite and Γ be a length-bounded EFS that is reduced with respect to X. Then for any $A \leftarrow B_1, \ldots, B_n \in \Gamma$, $|A| \leqq \max\{ |w|; w \in X \}$.

From Lemma 9 and the fact that there exist only finitely many patterns shorter than a fixed length except renaming of variables, we have the following.

LEMMA 10 For any finite set $X \subseteq \Sigma^+$ and any $n \geqq 0$,

$$\{ L(\Gamma) \mid \Gamma \text{ is length-bounded, } \#\Gamma \leqq n, \Gamma \text{ is reduced with respect to } X \}$$

consists of finitely many languages.

Proof. Let $X \subseteq \Sigma^+$ be finite, $l = \max\{|w|; w \in X\}$, Γ be a length-bounded EFS such that Γ is reduced with respect to X and $\#\Gamma \leqq n$, and $C = A \leftarrow B_1, \ldots, B_m \in \Gamma$. By Lemma 9, $|A| \leqq l$. Since C is length-bounded and $|B_i| \geqq 1$ for any $i = 1, \ldots, m$, $m \leqq l$. Therefore each clause in Γ contains at most $l+1$ distinct predicate symbols, so Γ contains at most $n \times (l+1)$ distinct predicate symbols. It appears that there exist only finitely many pairwise inequivalent length-bounded EFS's Γ such that $\#\Gamma \leqq n$ and $A \leftarrow B_1, \ldots, B_m \in \Gamma$ implies $|A| \leqq l$. From Lemma 6 the number of languages defined by such EFS's is finite. □

LEMMA 11 Let $U = \Sigma^+$, E be the set of all length-bounded clauses, and $M(\Gamma) = L(\Gamma, p_0)$. Then (U, E, M) is a concept defining framework that has bounded finite thickness.

From this lemma we have the following as a corollary of Theorem 5.

COROLLARY 12 For any $n \geqq 0$, the class of languages definable by length-bounded EFS's consisting of at most n clauses is inferable from positive data.

4.2 Context-Sensitive Grammars

Let Σ be a finite set of symbols as in the previous section and V be a set disjoint from Σ. An element in V is called a *nonterminal symbol*. We assume V contains a special nonterminal symbol S_0.

DEFINITION A *production* is an expression of the form $\alpha \rightarrow \beta$, where $\alpha, \beta \in (\Sigma \cup V)^+$. A *grammar* is a finite set of productions. A production $\alpha \rightarrow \beta$ is said to be *context-sensitive* if $|\alpha| \leqq |\beta|$. A *context-sensitive* grammar is a grammar whose productions are all context-sensitive.

DEFINITION Let Γ be a grammar. We define a binary relation $\Rightarrow_\Gamma$ on $(\Sigma \cup V)^+$ by

$$\gamma\alpha\delta \Rightarrow_\Gamma \gamma\beta\delta \quad \text{if } \alpha \rightarrow \beta \in \Gamma,$$

where $\alpha, \beta, \gamma, \delta \in (\Sigma \cup V)^+$. By $\Rightarrow_\Gamma^*$ we denote the reflexive transitive closure of $\Rightarrow_\Gamma$. The *language* $L(\Gamma)$ of a grammar Γ is defined by

$$L(\Gamma) = \{ w \in \Sigma^+ \mid S_0 \Rightarrow_\Gamma^* w \}.$$

Similar discussion as for length-bounded EFS's can be made for context-sensitive grammars. That is, it can be shown that (U, E, M) has bounded finite thickness, where $U = \Sigma^+$, E is the set of all context-sensitive productions, $M(\Gamma) = L(\Gamma)$. Thus we have the following corollary.

COROLLARY 13 For any n, the class of languages defined by context-sensitive grammar consisting of at most n productions is inferable from positive data.

4.3 Linear Prolog Programs

Let X, F and Π be mutually disjoint sets. An element in X, F or Π is called *variable*, *function symbol* or *predicate symbol*, respectively. We assume F and Π are finite. Each function or predicate symbol is associated with a nonnegative integer called *arity*. Function symbols with arity 0 are also called *constant symbols*.

DEFINITION A *term* is a variable, a constant symbol, or an expression of the form $f(t_1, \ldots, t_n)$, where f is a function symbol with arity $n \geqq 1$ and $t_1, \ldots, t_n$ are terms. A *ground term* is a term that does not contain any variable.

Atomic formulas (atoms), *ground atoms* and *definite clauses* are defined in a similar way to those for EFS's. A *program* is a finite set of definite clauses. *Substitutions*, *Herbrand base*, *minimal models* and other notions are defined in the ordinal ways [8].

DEFINITION The *length* of a term t, denoted by $|t|$ is defined inductively as follows:

(1) If t is a variable or a constant symbol, then $|t| = 1$.
(2) If $t = f(t_1, \ldots, t_n)$, then $|t| = |t_1| + \ldots + |t_n| + 1$.

For an atom $A = p(t_1, \ldots, t_n)$, we define $|A| = |t_1| + \ldots + |t_n|$.

DEFINITION A definite clause $A \rightarrow B_1, \ldots, B_n$ is *linear* if $|A\theta| \geqq |B_i\theta|$ for any substitution θ and any $i = 1, \ldots, n$. A program Γ is *linear* if clauses in Γ are all linear.

For linear programs almost the same discussions as for length-bounded EFS's can be done. From such discussions we can show that (U, E, M) has bounded finite thickness, where U is the Herbrand base, E is the set of all linear clauses, and $M(\Gamma)$ is the minimal model of a program Γ.

COROLLARY 14 For any $n \geqq 1$, the class of minimal models of linear programs consisting of at most n clauses is inferable from positive data.

If we construct an inference machine *IM* for linear programs based on Corollary 14, any example about every predicate symbol should be presented to *IM*, as in Model Inference System by Shapiro [10]. However, in some situation, examples about other predicate symbols than a special one might not be available. For example, identification of an EFS languages should work on words, which can tell nothing about predicate symbols but a special one p_0. Fortunately, even in such a situation, linear Prolog programs can be inferred from positive data.

Let p_0 be a special predicate symbol with arity m, T denotes the set of all ground terms. Further, let $U = T^m = \{(t_1, \ldots t_m) \mid t_i \in T\ (i=1, \ldots, m)\}$, E be the set of all linear

clauses, and $M'(\Gamma) = \{ \bar{t} \in T^m \mid p_0(\bar{t}) \in M(\Gamma) \}$. Then, we can show that a triple (U', E, M') is also a concept defining framework that has bounded finite thickness.

Acknowledgments

The author wishes to thank Dr. Dana Angluin for her invaluable comments that teaches him how to extend his results in the previous paper.

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] Motoki, T. and Shinohara, T.: Correct Definition of Finite Elasticity, Technical Report RIFIS-TR-CS-29, Kyushu University, 1990.

[10] Shapiro, E.Y.: Inductive Inference of Theories From Facts, YALEU / DCS / TR - 192, 1981.

[11] Shapiro, E.Y.: Alternation and the computational complexity of logic programs, J. Logic Program., 1, 19 - 33, 1984.

[12] Shinohara, T.: Polynomial time inference of extended regular pattern languages, LNCS, **147**, 115 - 127, 1983.

[13] Shinohara, T.: Inferring unions of two pattern languages, Bull. Inf. Cybern., **20**, 83 - 88, 1983.

[14] Shinohara, T.: Inductive inference of formal systems from positive data, Bull. Inf. Cybern., **22**, 9 - 18, 1986.

[15] Shinohara, T.: Inductive inference from positive data is powerful, Technical Report RIFIS-TR-CS-20, Kyushu University, 1989.

[16] Smullyan, R. M.: *Theory of Formal Systems,* Princeton Univ. Press, 1961.

[17] Wright, K.: Identification of unions of languages drawn from an identifiable class, Proc. 2nd Workshop Comput. Learning Theory, 328 - 333, 1989.

[18] Yamamoto, A.: Elementary formal system as a logic programming language, Proc. Logic Program. Conf. '89, ICOT, 123 - 132, 1989.