

Elementary Formal Systems, Intrinsic Complexity, and Procrastination

Sanjay Jain

Department of Information Systems and Computer Science
National University of Singapore
Singapore 119260, Republic of Singapore
Email: sanjay@iscs.nus.sg

Arun Sharma*

School of Computer Science and Engineering
The University of New South Wales
Sydney, NSW 2052, Australia
Email: arun@cse.unsw.edu.au

March 11, 2007

Abstract

Recently, rich subclasses of elementary formal systems (*EFS*) have been shown to be identifiable in the limit from only positive data. Examples of these classes are Angluin's pattern languages, unions of pattern languages by Wright and Shinohara, and classes of languages definable by length-bounded elementary formal systems studied by Shinohara. The present paper employs two distinct bodies of abstract studies in the inductive inference literature to analyze the learnability of these concrete classes.

The first approach, introduced by Freivalds and Smith, uses constructive ordinals to bound the number of mind changes. ω denotes the first limit ordinal. An ordinal mind change bound of ω means that identification can be carried out by a learner that after examining some element(s) of the language announces an upper bound on the number of mind changes it will make before converging; a bound of $\omega \cdot 2$ means that the learner reserves the right to revise this upper bound once; a bound of $\omega \cdot 3$ means the learner reserves the right to revise this upper bound twice, and so on. A bound of ω^2 means that identification can be carried out by a learner that announces an upper bound on the number of times it may revise its conjectured upper bound on the number of mind changes. It is shown in the present paper that the ordinal mind change complexity for identification of languages formed by unions of up to n pattern languages is ω^n . It is

*This research was partially supported by the Australian Research Council Grant A49600456.

also shown that this bound is essential. Similar results are also shown to hold for classes definable by length-bounded elementary formal systems with up to n clauses.

The second approach, studied by Freivalds, Kinber and Smith and by Jain and Sharma, employs reductions to study the intrinsic complexity of learnable classes. It is shown that the class of languages formed by taking unions of up to $n + 1$ pattern languages is a strictly more difficult learning problem than the class of languages formed by the union of up to n pattern languages. It is also shown that a similar hierarchy holds for the bound on the number of clauses in the case of languages definable by length-bounded *EFS*.

In addition to building bridges between three distinct areas of inductive inference, viz., learnability of *EFS* subclasses, ordinal mind change complexity, and intrinsic complexity, this paper also presents results that relate topological properties of learnable classes to that of intrinsic complexity and ordinal mind change complexity. For example, it is shown that a class that is complete according to the reductions for intrinsic complexity has infinite elasticity.

Since *EFS* languages and their learnability results have counterparts in traditional logic programming, the present paper demonstrates the possibility of using abstract results of inductive inference to gain insights into inductive logic programming.

1 Introduction

Arikawa [Ari70] adapted Smullyan's [Smu61] elementary formal systems (*EFS*) for investigation of formal languages. Later, Arikawa et al. [ASY92] showed that *EFS* can also be treated as a logic programming language. Recently various subclasses of *EFS* have been investigated in the context of learnability. It has been shown that rich classes can be identified in the limit from only positive data. These learning techniques have been applied to knowledge acquisition from amino acid sequences (see Arikawa et al. [AMS⁺92, ASMS91]). From a theoretical point of view, investigations of the learnability of subclasses of elementary formal systems are important because they yield corresponding results about the learnability of subclasses of logic programs. Arimura and Shinohara [AS94] have used the insight gained from the learnability of *EFS* subclasses to show that a class of linearly covering logic programs with local variables is identifiable in the limit from only positive data. More recently, Krishna-Rao [KR96] has established the learnability from only positive data of an even larger class of logic programs.

We consider three subclasses of elementary formal systems in the present paper. The smallest of these classes, the collection of pattern languages (*PATTERN*), was first introduced by Angluin [Ang80] who showed that this class can be identified in the limit from only positive data. Shinohara [Shi86] showed that the class of pattern languages is not closed under union and many rich concepts can be represented by unions of pattern languages. He also showed that the class of languages formed by union of up to 2 pattern

languages ($PATTERN^2$) is identifiable in the limit from only positive data. Later, Wright [Wri89] generalized this result to show that the classes of languages formed by unions of up to n pattern languages ($PATTERN^n$) can be identified in the limit from only positive data. Shinohara [Shi94] later showed that an even richer class, the classes of languages definable by length-bounded elementary formal systems with up to n clauses ($LBEFS^n$), is identifiable in the limit from only positive data. An interesting aspect of these results is that they have counterparts for the learnability of traditional logic programs, e.g., the learnability of the class $LBEFS^n$ implies the learnability of the class of minimal models of linear Prolog programs consisting of at most n definite clauses (see Shinohara [Shi91] and Arimura [Ari89]). In this respect, these results are also about inductive logic programming.

In the present paper we employ two distinct bodies of work in the inductive inference literature to analyze the learnability of the above classes. The first approach, introduced by Freivalds and Smith [FS92], involves the use of constructive ordinals to bound the number of mind changes before the onset of convergence. The second approach ([FKS95, JS94, JS95]) involves the use of reductions to study the intrinsic complexity of identifiable classes. Both these abstract studies have yielded elegant results. The present paper shows that they can be employed to gain an insight into the learnability of concrete classes. In this paper, we also present results about how certain topological properties of learnable classes (such as finite elasticity and infinite elasticity) are related to intrinsic complexity and to ordinal mind change complexity.

In the rest of this section we proceed as follows: In Section 1.1, we describe the learning model; in Section 1.2, we informally describe results about EFS and ordinal mind change complexity; in Section 1.3, we discuss results about EFS and intrinsic complexity; and in Section 1.4, we briefly mention some results on the connection between topological properties of language classes and intrinsic complexity. Formal treatment is presented in Sections 2 and 3.

1.1 Identification

Let N denote the set of natural numbers, $\{0, 1, 2, \dots\}$. We first define the notion of texts for languages.

Definition 1 (a) A text for a language L is a mapping T from N into $N \cup \{\#\}$ such that L is the set of natural numbers in the range of T .

(b) $\text{content}(T)$ denotes the set of natural numbers in the range of T . (Thus, the content of a text never includes $\#$.)

(c) The initial sequence of text T of length n is denoted $T[n]$.

(d) The set of all finite initial sequences of N and $\#$'s is denoted SEQ .

Intuitively, a text T for a language L is a presentation of elements of L (possibly repeated) and no non-elements of L ; #'s in the presentation may be thought of as modeling pauses in data input¹. It is easy to see that there exists a computable bijection between SEQ and N . Members of SEQ are inputs to machines that learn grammars (acceptors) for r.e. languages. We let σ and τ , with or without decorations², range over SEQ. Λ denotes the empty sequence. $\text{content}(\sigma)$ denotes the set of natural numbers in the range of σ .

Definition 2 *A language learning machine is an algorithmic mapping from SEQ into $N \cup \{?\}$.*

Here we interpret a conjecture of ? by a machine as “no guess at this moment”. This is useful to avoid biasing the number of mind changes of a machine. For this paper, we assume, without loss of generality, that $\sigma \subseteq \tau$ and $\mathbf{M}(\sigma) \neq ?$ implies $\mathbf{M}(\tau) \neq ?$.

\mathbf{M} denotes a typical variable for a language learning machine. We also fix an acceptable programming system and interpret the output of a language learning machine as the index of a program in this system. Then, a program conjectured by a machine in response to a finite initial sequence may be viewed as a candidate accepting grammar for the language being learned. We say that \mathbf{M} converges on text T to i (written: $\mathbf{M}(T)$ converges to i) just in case for all but finitely many n , $\mathbf{M}(T[n]) = i$. The following definition introduces Gold’s criterion for successful identification of languages.

Definition 3 [Gol67]

- (a) \mathbf{M} **TxtEx-identifies** a text T just in case $\mathbf{M}(T)$ converges to a grammar for $\text{content}(T)$.
- (b) \mathbf{M} **TxtEx-identifies** an r.e. language L (written: $L \in \mathbf{M}$) just in case \mathbf{M} **TxtEx-identifies** each text T for L .
- (c) **TxtEx** denotes all such collections \mathcal{L} of r.e. languages such that some machine **TxtEx-identifies** each language in \mathcal{L} .

1.2 EFS and Procrastination

Natural numbers have been used as counters for bounding the number of mind changes. However, such bounds do not take into account the scenario in which a learning machine, after examining an element of the language is in a position to issue a bound on the number of mind changes it will make before the onset of convergence. For example, consider the class $\text{COINIT} = \{L \mid (\exists n)[L = \{x \mid x \geq n\}]\}$. Intuitively, COINIT is the collection of languages that contain all natural numbers except a finite initial segment. Clearly, a learning machine that, at any given time, finds the minimum element in the data seen so far, say n , and

¹Note that the only text for the empty language is an infinite sequence of #'s.

²Decorations are subscripts, superscripts and the like.

emits a grammar for the language $\{x \mid x \geq n\}$ can easily be seen to identify *COINIT* in the limit from positive data. It is also easy to see that the class *COINIT* cannot be identified by any machine that is required to converge within a constant number of mind changes; however, the machine identifying *COINIT* can, after examining an element of the language, issue an upper bound on the number of mind changes. The class *PATTERN* has a similar property because any string in a pattern language yields a finite set of patterns that are candidate patterns for the language being learned. To model such scenarios Freivalds and Smith [FS93] introduced the use of constructive ordinals as mind change counters. We illustrate the idea with a few examples; the formal definition is presented later.

\mathbf{TxtEx}_α denotes the set of collections of languages that can be identified in the limit from texts with an ordinal mind change bound α . For $\alpha \prec \omega$, the notion coincides with the earlier notion of bounded mind change identification. For $\alpha = \omega$, \mathbf{TxtEx}_ω denotes learnable classes for which there exists a machine that, after examining some element(s) of the language, can announce an upper bound on the number of mind changes it will make before the onset of successful convergence. Both, *COINIT* and *PATTERN* are members of \mathbf{TxtEx}_ω .

Proceeding on, the class $\mathbf{TxtEx}_{\omega,2}$ contains classes for which there is a learning machine that after examining some element(s) of the language announces an upper bound on the number of mind changes, but reserves the right to revise this upper bound once. Similarly, in the case of $\mathbf{TxtEx}_{\omega,3}$, the machine reserves the right to revise its upper bound twice, and so on. $\mathbf{TxtEx}_{\omega,2}$ contains classes for which the machine announces an upper bound on the number of times it may revise its conjectured upper bound on the number of mind changes, and so on. The name “procrastination” derives from the ability of a learning machine to delay coming up with a correct upper bound on the number of mind changes.

We are able to derive the ordinal mind change complexity of the classes of languages formed by taking unions of pattern languages. For $n > 1$, we show that the class formed by taking unions of up to n pattern languages, $PATTERN^n$, is in $\mathbf{TxtEx}_{\omega^n}$. We also show that there are cases for which the ω^n bound is essential because $PATTERN^n \notin \mathbf{TxtEx}_\alpha$, for all $\alpha \prec \omega^n$.³

We also consider the ordinal mind change complexity of languages definable by length bounded elementary formal systems. Like in the case of unions of pattern languages, we are able to show that the class of languages definable by length bounded *EFS* with no more than n clauses, $LBEFS^n$, is in $\mathbf{TxtEx}_{\omega^n}$. Since, the class of unions of n pattern languages is contained in $LBEFS^n$, it immediately follows that $LBEFS^n \notin \mathbf{TxtEx}_\alpha$, for all $\alpha \prec \omega^n$.

³Interestingly, if we consider languages formed by unions of pattern languages such that the patterns in the union have disjoint alphabet, then it is easy to verify that the ordinal mind change complexity turns out to be $\omega \cdot n$ for language classes containing unions of up to n pattern languages.

1.3 *EFS* and Intrinsic Complexity

Recently, a new approach to the study of “intrinsic” complexity of learning has been proposed for identification in the limit of functions by Freivalds, Kinber, and Smith [FKS95] and for identification in the limit of languages by Jain and Sharma [JS94, JS96a].

The main idea of the approach is to introduce reductions between learnable classes of languages. If a collection of languages, \mathcal{L}_1 , can be reduced to another collection of languages, \mathcal{L}_2 , then the learnability of \mathcal{L}_1 is no more difficult than that of \mathcal{L}_2 . Moreover, an algorithm for learning \mathcal{L}_2 can be transformed into an algorithm for learning \mathcal{L}_1 . Based on these reductions, one can define the notion of hardness and completeness. These reductions were used to show that the following three collections of languages, each of which is identifiable, pose learning problems of increasing difficulty: (a) *SINGLE*, the collection of singleton languages; (b) *COINIT*; and (c) *FIN*, the collection of finite languages. It was shown that *SINGLE* is reducible to *COINIT* but *COINIT* is not reducible to *SINGLE* and *COINIT* is reducible to *FIN* but *FIN* is not reducible to *COINIT*. It was discussed in [JS94] that this reduction captures the intuitive difficulty of learning these classes. *SINGLE* can be identified by a learning machine that can confirm its success. *COINIT* cannot be identified by any machine that can confirm its success, but can be identified by a machine, that after inspecting an element of the language, provides an upper bound on the number of mind changes. *FIN*, on the other hand, can neither be identified by a machine that confirms its success nor can it be learned by a machine that provides an upper bound on the number of mind changes after inspecting an element of the language. In fact according to a version of the reduction, *FIN* is complete—it poses the most difficult learning problem. It was also shown that the class *COINIT* is equivalent to *PATTERN*.

In the present paper, we investigate the intrinsic complexity for unions of pattern languages and languages definable by length-bounded *EFS*. Now, as mentioned above, since the ordinal mind change complexity of *PATTERN*² is ω^2 and that of *PATTERN* is ω , it appears that *PATTERN*² is a more difficult learning problem than *PATTERN*. Once again, the notion of intrinsic complexity captures this gradation in learning difficulty as can be seen from the following summary of our results. Let $n > 0$.

- (a) *PATTERN*ⁿ reducible to *PATTERN*ⁿ⁺¹ but *PATTERN*ⁿ⁺¹ not reducible to *PATTERN*ⁿ.
- (b) *LBEFS*ⁿ reducible to *LBEFS*ⁿ⁺¹ but *LBEFS*ⁿ⁺¹ not reducible to *LBEFS*ⁿ.

1.4 Topological Properties of Learnable Classes

Certain topological properties of language classes have been shown to be sufficient for identification in the limit of indexed families of recursive languages from positive data. In this

paper, we also present some results that relate these topological properties to the notion of intrinsic complexity and ordinal mind change complexity.

For example, Angluin [Ang80] defined a class \mathcal{L} to have *finite thickness* just in case for each $n \in N$, the cardinality of $\{L \in \mathcal{L} \mid n \in L\}$ is finite. She showed that if an indexed family of recursive languages \mathcal{L} has finite thickness then $\mathcal{L} \in \mathbf{TxtEx}$. It can be shown that if a class has finite thickness, then it is not complete in terms of any of the reductions for intrinsic complexity discussed in this paper.

Wright [Wri89] (see also Motoki, Shinohara, and Wright [MSW91]) introduced the notions of finite elasticity and infinite elasticity (to be defined later). Wright [Wri89] showed that if a class \mathcal{L} has finite thickness then it has finite elasticity. He also showed that if an indexed family of recursive languages \mathcal{L} has finite elasticity then $\mathcal{L} \in \mathbf{TxtEx}$. We can show that if a class \mathcal{L} is complete according to any of the reductions for intrinsic complexity discussed in this paper, then it has infinite elasticity. We also present some results about the relationship between finite elasticity and ordinal mind change complexity.

We now proceed formally. In Section 2, we introduce the notation and give formal definitions of *EFS*, intrinsic complexity, and procrastination. Results are presented in Section 3.

2 Preliminaries

N^+ denotes the set of positive integers. Any unexplained recursion theoretic notation is from [Rog67]. Cardinality of a set S is denoted by $\text{card}(S)$. \emptyset denotes the empty set. The maximum and minimum of a set are denoted by $\max(\cdot), \min(\cdot)$, respectively, where $\max(\emptyset) = 0$ and $\min(\emptyset) = \infty$. We let $\langle \cdot, \cdot \rangle$ stand for an arbitrary, computable, bijective mapping from $N \times N$ onto N [Rog67]. Similarly, one can define $\langle \cdot, \dots, \cdot \rangle$ for encoding multiple tuples of natural numbers onto N .

Symbol \mathcal{R} denotes the set of all total computable functions. By φ we denote a fixed *acceptable* programming system for the partial computable functions: $N \rightarrow N$ [Rog67, MY78]. By φ_i we denote the partial computable function computed by the program with number i in the φ -system.

By Φ we denote an arbitrary fixed Blum complexity measure [Blu67, HU79] for the φ -system. By W_i we denote $\text{domain}(\varphi_i)$. W_i is, then, the r.e. set/language ($\subseteq N$) accepted (or equivalently, generated) by the φ -program i . We also say that i is a grammar for W_i . Symbol \mathcal{E} will denote the set of all r.e. languages. Symbols \mathcal{L} and \mathcal{S} range over subsets of \mathcal{E} . We denote by $W_{i,s}$ the set $\{x \leq s \mid \Phi_i(x) < s\}$.

2.1 Procrastination: Ordinals as Mind Change Counters

We assume a fixed notation system and partial ordering of constructive ordinals as used by, for example, Kleene [Kle38, Rog67, Sac90]. \preceq, \prec, \succeq and \succ on ordinals below refer

to the partial ordering of ordinals which is provable in the notation system used. We do not go into the details of the notation system used, but instead refer the reader to [Kle38, Rog67, Sac90, CJS95, FS93].

Definition 4 \mathbf{F} , a mapping from SEQ into constructive ordinals, is an *ordinal mind change counter* just in case $(\forall \sigma \subseteq \tau)[\mathbf{F}(\sigma) \succeq \mathbf{F}(\tau)]$.

Definition 5 [FS93] Let α be a constructive ordinal.

- (a) We say that \mathbf{M} , with associated ordinal mind change counter function \mathbf{F} , \mathbf{TxtEx}_α -*identifies* a text T just in case the following three conditions hold:
 - (i) $\mathbf{M}(T)$ converges to a grammar for $\text{content}(T)$,
 - (ii) $\mathbf{F}(\Lambda) = \alpha$ and
 - (iii) $(\forall n)[? \neq \mathbf{M}(T[n]) \neq \mathbf{M}(T[n+1]) \Rightarrow \mathbf{F}(T[n]) \succ \mathbf{F}(T[n+1])]$.
- (b) \mathbf{M} , with associated ordinal mind change counter function \mathbf{F} , \mathbf{TxtEx}_α -*identifies* L (written: $L \in \mathbf{TxtEx}_\alpha(\mathbf{M}, \mathbf{F})$) just in case \mathbf{M} , with associated ordinal mind change counter function \mathbf{F} , \mathbf{TxtEx}_α -*identifies* each text for L .
- (c) $\mathbf{TxtEx}_\alpha = \{\mathcal{L} \mid (\exists \mathbf{M}, \mathbf{F})[\mathcal{L} \subseteq \mathbf{TxtEx}_\alpha(\mathbf{M}, \mathbf{F})]\}$.

2.2 Preliminaries of Intrinsic Complexity

We write “ $\sigma \subseteq \tau$ ($\sigma \subset \tau$)” if σ is an initial sequence (proper initial sequence) of τ . Likewise, we write $\sigma \subset T$ if σ is an initial finite sequence of text T . Let finite sequences $\sigma^0, \sigma^1, \sigma^2, \dots$ be given such that $\sigma^0 \subseteq \sigma^1 \subseteq \sigma^2 \subseteq \dots$ and $\lim_{i \rightarrow \infty} |\sigma^i| = \infty$. Then there is a unique text T such that for all $n \in \mathbb{N}$, $\sigma^n \subset T$. This text is denoted $\bigcup_n \sigma^n$. Let \mathbf{TEXTS} denote the set of all texts, that is, the set of all infinite sequences over $\mathbb{N} \cup \{\#\}$.

Definition 6 An *enumeration operator*, Θ , is an algorithmic mapping from SEQ into SEQ such that the following two conditions are satisfied

- (a) for all $\sigma, \tau \in \text{SEQ}$, if $\sigma \subseteq \tau$, then $\Theta(\sigma) \subseteq \Theta(\tau)$;
- (b) For all texts T , $\lim_{n \rightarrow \infty} |\Theta(T[n])| = \infty$.

By extension, we think of Θ as also defining a mapping from \mathbf{TEXTS} into \mathbf{TEXTS} such that $\Theta(T) = \bigcup_n \Theta(T[n])$. Furthermore we define $\Theta(L) = \{\text{content}(\Theta(T)) \mid T \text{ is a text for } L\}$. Intuitively, $\Theta(L)$ denotes the set of languages to whose texts Θ maps texts of L . The reader should note the overloading of this notation because the type of the argument to Θ could be a sequence, a text, or a language; it will be clear from the context which usage is intended. We let Θ and Ψ , with or without decorations, range over enumeration operators. Since we

will only be dealing with enumeration operators, we often drop the prefix “enumeration”. Finally, we say that a sequence of grammars $G = g_0, g_1, g_2, \dots$ is **TxtEx**-admissible for a text T just in case this sequence of grammars converges to a grammar for $\text{content}(T)$. We now introduce our first reduction. The reader is referred to [JS94] for a detailed discussion.

Definition 7 Let $\mathcal{L} \subseteq \mathcal{E}$ and $\mathcal{L}' \subseteq \mathcal{E}$ be given. Let $\mathcal{T} = \{T \mid (\exists L \in \mathcal{L})[T \text{ is a text for } L]\}$. Let $\mathcal{T}' = \{T \mid (\exists L \in \mathcal{L}') [T \text{ is a text for } L]\}$. We say that $\mathcal{L} \leq_{\text{weak}}^{\text{TxtEx}} \mathcal{L}'$ just in case there exist operators Θ and Ψ such that for all $T \in \mathcal{T}$ and for all infinite sequence of grammars $G = g_0, g_1, \dots$, the following two conditions hold:

- (a) $\Theta(T) \in \mathcal{T}'$ and
- (b) if G is a **TxtEx**-admissible sequence for $\Theta(T)$, then $\Psi(G)$ is a **TxtEx**-admissible sequence for T .

In the above case, we also say that Θ and Ψ witness $\mathcal{L} \leq_{\text{weak}}^{\text{TxtEx}} \mathcal{L}'$. We say that $\mathcal{L} \equiv_{\text{weak}}^{\text{TxtEx}} \mathcal{L}'$ iff $\mathcal{L} \leq_{\text{weak}}^{\text{TxtEx}} \mathcal{L}'$ and $\mathcal{L}' \leq_{\text{weak}}^{\text{TxtEx}} \mathcal{L}$.

The next definition describes the notions of hardness and completeness for the above reduction.

Definition 8 Let $\mathcal{L} \subseteq \mathcal{E}$ be given.

- (a) We say that \mathcal{L} is $\leq_{\text{weak}}^{\text{TxtEx}}$ -hard iff for all $\mathcal{L}' \in \mathbf{TxtEx}$, $\mathcal{L}' \leq_{\text{weak}}^{\text{TxtEx}} \mathcal{L}$.
- (b) We say that \mathcal{L} is $\leq_{\text{weak}}^{\text{TxtEx}}$ -complete iff \mathcal{L} is $\leq_{\text{weak}}^{\text{TxtEx}}$ -hard and $\mathcal{L} \in \mathbf{TxtEx}$.

It should be noted that, in Definition 7, there is no requirement that Θ map every text for a language in \mathcal{L} into texts for a unique language in \mathcal{L}' . If we further place such a constraint on Θ , we get the following stronger notion.

Definition 9 Let $\mathcal{L} \subseteq \mathcal{E}$ and $\mathcal{L}' \subseteq \mathcal{E}$ be given. We say that $\mathcal{L} \leq_{\text{strong}}^{\text{TxtEx}} \mathcal{L}'$ just in case there exist operators Θ, Ψ such that

- (a) Θ and Ψ witness that $\mathcal{L} \leq_{\text{weak}}^{\text{TxtEx}} \mathcal{L}'$, and
- (b) for all $L \in \mathcal{L}$, $\Theta(L)$ contains exactly one language.

In other words, for all $L \in \mathcal{L}$, there exists an $L' \in \mathcal{L}'$, such that $(\forall \text{ texts } T \text{ for } L)[\Theta(T) \text{ is a text for } L']$. In the above case, we also say that Θ and Ψ witness $\mathcal{L} \leq_{\text{strong}}^{\text{TxtEx}} \mathcal{L}'$. We say that $\mathcal{L} \equiv_{\text{strong}}^{\text{TxtEx}} \mathcal{L}'$ iff $\mathcal{L} \leq_{\text{strong}}^{\text{TxtEx}} \mathcal{L}'$ and $\mathcal{L}' \leq_{\text{strong}}^{\text{TxtEx}} \mathcal{L}$.

We can similarly define $\leq_{\text{strong}}^{\text{TxtEx}}$ -hardness and $\leq_{\text{strong}}^{\text{TxtEx}}$ -completeness.

2.3 Elementary Formal Systems

Let Σ , X , and Π be mutually disjoint sets. Σ and Π are finite and their elements are referred to as *constant symbols* and as *predicate symbols*, respectively. Elements of X are referred to as *variables*. We let a, b, \dots range over constant symbols; x, y, z, x_1, x_2, \dots range over variables; and p, q, p_1, p_2, \dots range over predicate symbols. Each predicate symbol is associated with a nonnegative integer called its *arity*.

Definition 10 A *term* or a *pattern* is an element of $(\Sigma \cup X)^+$. A *ground term* (or a *word*, or a *string*) is an element of Σ^+ .

A *substitution* is a homomorphism from terms to terms that maps each symbol $a \in \Sigma$ to itself. The image of a term π under a substitution θ is denoted $\pi\theta$. We next describe the language defined by a pattern. Note that there exists a recursive bijective mapping between elements of Σ^+ and N . Thus we can identify elements of Σ^+ with elements of N . We implicitly assume such an identification when we discuss languages defined using subsets of Σ^+ below. (We do not explicitly use the bijective mapping for ease of notation).

Definition 11 [Ang80] The language associated with the pattern π is defined as $\mathbf{Lang}(\pi) = \{\pi\theta \mid \theta \text{ is a substitution and } \pi\theta \in \Sigma^+\}$. We define the class $PATTERN = \{\mathbf{Lang}(\pi) \mid \pi \text{ is a pattern}\}$.

Angluin [Ang80] showed that $PATTERN \in \mathbf{TxtEx}$. Shinohara [Shi86] showed that pattern languages are not closed under union, and hence it is useful to study identification of languages that are unions of more than one pattern language, as they can be used to represent more expressive concepts.

We next define unions of pattern languages. Let S be a set of patterns. Then $\mathbf{Lang}(S)$ is defined as $\bigcup_{\pi \in S} \mathbf{Lang}(\pi)$. Intuitively, $\mathbf{Lang}(S)$ is the language formed by the union of languages associated with patterns in S .

Definition 12 [Shi86, Wri89] Let $n > 1$. $PATTERN^n = \{\mathbf{Lang}(S) \mid \text{card}(S) \leq n\}$.

Shinohara [Shi86] and Wright [Wri89] showed that for $n > 1$, $PATTERN^n \in \mathbf{TxtEx}$.

We next consider languages definable by length-bounded elementary formal systems. For this we first define atoms, definite clauses, and elementary formal systems.

Definition 13 [ASY92]

- (a) An *atomic formula* (or, an *atom*) is an expression of the form $p(\pi_1, \pi_2, \dots, \pi_n)$, where the arity of $p \in \Pi$ is n and $\pi_1, \pi_2, \dots, \pi_n$ are terms. An atom $p(\pi_1, \pi_2, \dots, \pi_n)$ is ground if terms $\pi_1, \pi_2, \dots, \pi_n$ are all ground.

- (b) 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, B_n are atoms. A is called the *head* of the clause and the sequence of atoms B_1, \dots, B_n is called the *body* of the clause.
- (c) An *elementary formal system* (EFS) is a finite set of definite clauses, which are called axioms.

Let $p(\pi_1, \dots, \pi_n)$ be an atom and let $C = A \leftarrow B_1, \dots, B_n$ be a clause. Then $p(\pi_1, \dots, \pi_n)\theta$ is defined as $p(\pi_1\theta, \dots, \pi_n\theta)$ and $C\theta$ is defined as $A\theta \leftarrow B_1\theta, \dots, B_n\theta$. We next define what it means for a clause to be provable from an EFS.

Definition 14 [ASY92] A clause C is provable from an EFS Γ (written: $\Gamma \vdash C$) just in case C is obtained from Γ by a finite number of applications of substitutions and modus ponens.

We next define what it means for a language definable by an EFS.

Definition 15 [ASY92] Let Γ be an EFS and let p be a predicate symbol with arity n . Then $\mathbf{Lang}(\Gamma, p)$ is defined to be $\{(w_1, \dots, w_n) \in (\Sigma^+)^n \mid \Gamma \vdash p(w_1, \dots, w_n)\}$. If p is unary then $\mathbf{Lang}(\Gamma, p)$ is a language over Σ . A language L is *definable by EFS* or is an *EFS language* just in case there exist Γ and p such that $\mathbf{Lang}(\Gamma, p) = L$.

$|\pi|$ denotes the length of a term π ; the length of an atom $p(\pi_1, \dots, \pi_n)$ is defined as follows: $|p(\pi_1, \dots, \pi_n)| = |\pi_1| + \dots + |\pi_n|$. We are interested in languages definable by length bounded elementary formal systems.

Definition 16 [ASY92, Shi94]

- (a) A clause $A \leftarrow B_1, \dots, B_n$ is *length-bounded* just in case $|A\theta| \geq |B_1\theta| + \dots + |B_n\theta|$ for any substitution θ .
- (b) An EFS Γ is *length-bounded* just in case each axiom of Γ is length-bounded.
- (c) For $n \geq 1$, $\mathbf{LBEFS}^n = \{\mathbf{Lang}(\Gamma, p) \mid \Gamma \text{ is length-bounded, } p \text{ is unary, } \text{card}(\Gamma) \leq n, \mathbf{Lang}(\Gamma, p) \neq \emptyset\}$.

Shinohara [Shi94] showed that for $n \geq 1$, $\mathbf{LBEFS}^n \in \mathbf{TxtEx}$.

3 Results

3.1 EFS and Ordinal Mind Change Complexity

In this section we show that the idea of bounding the number of mind changes by constructive ordinals can be used to analyze the mind change complexity of many concrete classes of languages. It is easy to verify that $\mathbf{PATTERN} \in \mathbf{TxtEx}_\omega$. This means that $\mathbf{PATTERN}$

is identifiable by a learning machine that, after examining an element of the language being identified, can issue an upper bound on the number of mind changes it will make before converging.

We first consider the identification of unions of pattern languages. The following theorem says that the ordinal mind change complexity for identification of unions of up to d pattern languages is ω^d . Later, Corollary 1 establishes that there are cases for which ω^d bound is essential because $PATTERN^d \notin \mathbf{TxtEx}_\alpha$, for all $\alpha \prec \omega^d$.

Theorem 1 *For $d \in \mathbb{N}^+$, $PATTERN^d \in \mathbf{TxtEx}_{\omega^d}$.*

The proof of the above theorem is facilitated by two lemmas, which we present next.

Lemma 1 *There exists a recursive function f (from finite sets to \mathbb{N}) such that the following holds. Suppose S is a nonempty finite set of decision procedures and T is a text. Then $\mathbf{M}_{f(S)}$ satisfies the following properties.*

- (A) $(\forall n)[\mathbf{M}_{f(S)}(T[n]) \in S]$;
- (B) *the number of mind changes by $\mathbf{M}_{f(S)}$ on T is bounded by $2 * \text{card}(S)$; and*
- (C) *if there exists a decision procedure for $\text{content}(T)$ in S , then $\mathbf{M}_{f(S)}(T)$ converges to a decision procedure for $\text{content}(T)$.*

PROOF. Suppose S , a finite set of decision procedures, is given. We will describe $\mathbf{M}_{f(S)}$. It will be easy to see that the description of $\mathbf{M}_{f(S)}$ is effective in S . (Note that if S is not a set of decision procedures, then we do not care about the behavior of $\mathbf{M}_{f(S)}$).

For $i \in S$, let L_i denote the language accepted by the decision procedure i . Let $L_i[n]$ denote $\{x \in L_i \mid x < n\}$.

Begin $\mathbf{M}_{f(S)}(T[n])$

1. Let $X_n = \{i \in S \mid \text{content}(T[n]) \subseteq L_i\}$.
 2. **if** there exists an $i \in X_n$, such that $(\forall i' \in X_n)[L_i[n] \subseteq L_{i'}[n]]$
 - 2a. **then** output least such i (say i_n)
 - 2b. **else** output $\mathbf{M}_{f(S)}(T[n-1])$.
- endif**

End $\mathbf{M}_{f(S)}(T[n])$

It is easy to observe that \mathbf{M} satisfies (A). We now show (C). Suppose i is the least decision procedure in S for $\text{content}(T)$. Let $X = \{i' \in S \mid \text{content}(T) \subseteq L_{i'}\}$. It is easy to observe that $\lim_{n \rightarrow \infty} X_n = X$. Let n be large enough so that $X_n = X$. For such n , the **if** clause in Step 2 succeeds, $\mathbf{M}_{f(S)}(T[n]) = i$. Thus (C) holds.

We now show that (B) holds. First note that $X_n \supseteq X_{n+1}$. Note that the only interesting steps for counting mind changes are those in which the **if** condition in Step 2 succeeds and Step 2a is executed.

Claim 1 Suppose that, for $n < n' < n''$, step 2 succeeds, and $i_n = i_{n''}$ but $i_n \neq i_{n'}$. Then $i_{n'} \notin X_{n''}$.

PROOF. Suppose the hypothesis. Then, $\{i_n, i_{n'}\} \subseteq X_{n'}$. Thus, $L_{i_{n'}}[n'] \subseteq L_{i_n}[n']$ and hence $L_{i_{n'}}[n] \subseteq L_{i_n}[n]$. Since i_n was chosen over $i_{n'}$ in $\mathbf{M}_{f(S)}(T[n])$, it follows that $i_n < i_{n'}$. However, since $i_{n'}$ was chosen (and thus i_n was not chosen) in step 2a of $\mathbf{M}_{f(S)}(T[n'])$, it follows that $L_{i_n}[n'] \not\subseteq L_{i_{n'}}[n']$. Hence, $L_{i_n}[n''] \not\subseteq L_{i_{n'}}[n'']$. Thus since $i_{n''} = i_n$, it follows that $i_{n'} \notin X_{n''}$. \square

We say that a sequence of hypothesis of \mathbf{M} is *stacklike* if it can be obtained by doing push/pop/top operations on a stack, where each element of S is pushed on the stack at most once. It immediately follows from the above claim that the sequence of hypothesis output by \mathbf{M} is stacklike. Also it is easy to see that in any stacklike sequence, the number of mind changes is bounded by $2 * \text{card}(S)$. Clause (B) and the Lemma follows. \blacksquare

Note that for the above lemma, the grammar can be of any form as long as the membership in each of them is decidable. Thus they could be pattern languages or unions of pattern languages, etc.

The next lemma requires the development of some technical machinery.

Definition 17 A *Search Tree* is a finite labeled rooted tree. We denote the label of node, v , in search tree H by $C_H(v)$.

Intuitively, the label on the nodes are interpreted as decision procedures. We abuse the notation slightly and by $\mathbf{Lang}(C_H(v))$, we mean the language decided by $C_H(v)$. We next introduce a partial order on search trees.

Definition 18 Suppose H_1 and H_2 are two search trees. We say that $H_1 \preceq H_2$ just in case the following properties are satisfied:

- (A) root of H_1 has the same label as root of H_2 ;
- (B) H_1 is a labeled subgraph of H_2 ; and
- (C) all nodes of H_1 , except the leaves, have exactly the same children in both H_1 and H_2 .

Essentially, $H_1 \preceq H_2$ means that H_2 is obtained by attaching some (possibly zero) trees to some of the leaves of the search tree H_1 . It is helpful to formalize the notion of *depth* of a search tree as follows: depth of root is 0; depth of a child is 1 + depth of parent; and depth of a search tree is depth of the deepest leaf.

Q , a mapping from SEQ to search trees, is called a *d-Explorer* iff the following properties are satisfied:

- (A) $\sigma \subseteq \tau \Rightarrow Q(\sigma) \preceq Q(\tau)$;
- (B) $(\forall \sigma)[\text{depth}(Q(\sigma)) \leq d]$; and

(C) for all T , $Q(T) \downarrow$, i.e., $(\forall n) [Q(T[n]) = Q(T[n+1])]$.

(The reader should note that C is actually implied by A and B; C has been included to emphasize the point.)

Lemma 2 *Suppose Q is a d -Explorer. Then there exists a machine \mathbf{M} and an associated ordinal mind change counter \mathbf{F} such that the following properties are satisfied:*

(A) $(\forall T)[\mathbf{M}(T) \downarrow]$;

(B) $\mathbf{F}(\Lambda) = \omega^d$; and

(C) *if there exists a node v in $Q(T)$ such that $C_{Q(T)}(v)$ is a decision procedure for $\text{content}(T)$, then \mathbf{M} , with associated mind change counter \mathbf{F} , $\mathbf{TextEx}_{\omega^d}$ -identifies T .*

PROOF. The idea is to use the set of labels on the search trees generated by Q along with Lemma 1. In fact, for clause (C), \mathbf{M} constructed below will converge to a decision procedure for $\text{content}(T)$ (which can be easily converted to a grammar if needed).

Let f be as in Lemma 1. $\mathbf{M}(T[n]) = \mathbf{M}_{f(S_n)}(T[n])$, where S_n denotes the set of decision procedures which are labels of nodes in $Q(T[n])$. For the mind change function \mathbf{F} we do the following. First we associate an ordinal with each search tree H .

If H is just a root then $\text{Ordval}(H) = \omega^d$; otherwise, $\text{Ordval}(H) = \sum_{i=1}^d [\omega^{d-i} \cdot k_i] + c$, where k_i denotes the number of leaves in H at depth i , and c is $2 \cdot (\text{number of nodes in } H) + 2$.

We define $\mathbf{F}(T[n])$ as follows. Suppose, $n' \leq n$ is the largest number such that $Q(T[n']) = Q(T[n])$ (i.e., the last time Q made a mind change). Let c' denote the number of mind changes made by \mathbf{M} , between $T[n']$ and $T[n]$, i.e., $c' = \text{card}(\{n'' \mid n' \leq n'' < n \wedge \mathbf{M}(T[n'']) \neq \mathbf{M}(T[n''+1])\})$. Then let $F(T[n]) = \text{Ordval}(Q(T[n])) - c'$. (It is easy to verify, using Lemma 1, that this subtraction is well defined.)

It is also easy to verify using Lemma 1 that value of $\mathbf{F}(T[n]) \geq \text{Ordval}(Q(T[n])) - c$, where c is $2 \cdot (\text{number of nodes in } H) + 1$. The reader can easily observe that \mathbf{F} is indeed an ordinal mind change counter. Now, using the convergence of Q , the definition of \mathbf{M} , and Lemma 1, one can immediately verify the clauses (A), (B) and (C) of the lemma. ■

PROOF. (Theorem 1) Using Lemma 2 we just need to show the existence of a d -Explorer Q with “nice” properties. For ease of presentation, we will use sets of pattern languages as labels for nodes of search trees. It is easy to convert these labels into decision procedures.

We construct a d -explorer Q as follows.

Let $Q(\Lambda) =$ just a root with label \emptyset .

$Q(T[n+1])$ is obtained as follows.

For each leaf, v , in $Q(T[n])$, such that $\text{depth}(v) < d$ and $\text{content}(T[n+1]) \not\subseteq \mathbf{Lang}(C_H(v))$ do the following:

For each pattern π , such that length of π is bounded by the maximum length string in $\text{content}(T[n+1])$, add a child with label $[C_{H(v)} \cup \{\pi\}]$ to v .

It is easy to verify that Q is d -Explorer. Moreover, for any text T for a language $L \in \text{PATTERN}^d$, there exists a leaf, v , in $Q(T)$, such that $\mathbf{Lang}(C_{Q(T)}(v)) = L$. ■

The following corollary to Theorem 4 (presented in the Section 3.2) shows that the bound of ω^d for identification of PATTERN^d is essential.

Corollary 1 *For each $d > 0$, $\text{PATTERN}^d \notin \bigcup_{\alpha < \omega^d} \mathbf{TxtEx}_\alpha$.*

Note that in the proof of Theorem 1 we constructed d -Explorers using the following property of pattern languages: strings of length n can only be generated by patterns of length $\leq n$. LBEFS have a similar property: proof of an atom of length n , can only use clauses whose head have length bounded by n (and hence the length of the clauses are bounded).

Thus, another application of Lemma 2 can be used to show ω^d is also the ordinal mind change complexity of the language class definable by length-bounded elementary formal systems with at most d axioms.

Theorem 2 *Let $d > 0$. $\text{LBEFS}^d \in \mathbf{TxtEx}_{\omega^d}$.*

Since for each $d > 0$, $\text{PATTERN}^d \subset \text{LBEFS}^d$, the following corollary immediately follows from Corollary 1.

Corollary 2 *For each $d > 0$, $\text{LBEFS}^d \notin \bigcup_{\alpha < \omega^d} \mathbf{TxtEx}_\alpha$.*

3.2 EFS Subclasses and Intrinsic Complexity

In this section, we compare the intrinsic complexity of unions of pattern languages and languages of bounded cardinality.

In [JS94, JS96a] it was shown that PATTERN is a strictly more difficult learning problem than the class of singleton languages, i.e., $\text{SINGLE} \leq_{\text{strong}}^{\mathbf{TxtEx}} \text{PATTERN}$, but $\text{PATTERN} \not\leq_{\text{weak}}^{\mathbf{TxtEx}} \text{SINGLE}$. Let $\text{FIN}_i = \{S \mid 1 \leq \text{card}(S) \leq i\}$. It has also been shown that PATTERN and FIN_i , $i > 1$, are incomparable. In this section we relate the intrinsic complexity of FIN_i , $i > 1$, to the intrinsic complexity of unions of pattern languages. It is easy to establish the following proposition.

Proposition 1 $(\forall i \in \mathbb{N}^+)[\text{FIN}_i \leq_{\text{strong}}^{\mathbf{TxtEx}} \text{PATTERN}^i]$.

Surprisingly, the above result can be strengthened to show the following theorem which says that the classes of finite languages of bounded cardinality are no more difficult to learn than class of languages formed by unions of up to two pattern languages.

Theorem 3 $(\forall k \in \mathbb{N}^+)[\text{FIN}_k \leq_{\text{strong}}^{\mathbf{TxtEx}} \text{PATTERN}^2]$.

PROOF. We give the mappings Θ and Ψ witnessing the theorem.

Suppose $y_1 < y_2 < \dots < y_r$, and $r \leq k$. Let

$$\Theta(\{y_1, y_2, \dots, y_r\}) = \mathbf{Lang}(a^{2(k+1-r)}x) \cup \mathbf{Lang}(a^{2(k+1-r)-1}bb^{\langle y_1, y_2, \dots, y_r \rangle}).$$

Note that it is easy to construct such a Θ .

Suppose f is a recursive function from finite sets to natural numbers such that $f(\{y_1, y_2, \dots, y_r\})$ is a grammar for $\{y_1, y_2, \dots, y_r\}$. Let $\Psi(g_1, g_2, \dots, g_l) = (g'_1, g'_2, \dots, g'_l)$, where g'_i , $1 \leq i \leq l$, is defined as follows:

$$g'_i = \begin{cases} f(\{y_1, y_2, \dots, y_r\}), & \text{if } a^{2(k+1-r)-1}bb^{\langle y_1, y_2, \dots, y_r \rangle} \in W_{g_i, i} \text{ and} \\ & r = \max(\{r' \mid (\exists x \in \Sigma^*) a^{2(k+1-r')-1}bx \in W_{g_i, i}\}); \\ 0, & \text{otherwise.} \end{cases}$$

It is easy to verify that Θ, Ψ witness $FIN_k \leq_{\text{strong}}^{\mathbf{TxE}} PATTERN^2$. ■

An immediate consequence of this result is the following corollary.

Corollary 3 $PATTERN^2 \not\leq_{\text{weak}}^{\mathbf{TxE}} PATTERN$.

We next present an interesting result which shows that $PATTERN^{d+1}$ is not reducible to any class that can be learned with an ordinal mind change complexity of α , $\alpha \prec \omega^{d+1}$. We have already noted that this result implies Corollary 1.

Theorem 4 *Let $d \in N^+$, $\alpha \prec \omega^{d+1}$. Suppose $\mathcal{L} \in \mathbf{TxE}_\alpha$. Then $PATTERN^{d+1} \not\leq_{\text{weak}}^{\mathbf{TxE}} \mathcal{L}$.*

PROOF. We first prove a lemma regarding existence of pattern languages with certain properties. For ordinals used in the lemma and the rest of the proof of the theorem we are concerned with the values of the ordinals and not about the exact notation. Thus for the purpose of diagonalization, all notation for the same ordinal will be equivalent. The equality/precedence of ordinals in this proof are based on the values of the ordinals and not on the exact notation.

Lemma 3 *Let $d \in N^+$. Then, for each $\beta \prec \omega^{d+1}$, we can define pattern languages L_β^1, L_β^2 such that the following property holds:*

$$(\forall \beta' \prec \beta \prec \omega^{d+1}) [L_\beta^1 \subset L_\beta^2 \subset L_{\beta'}^1].$$

PROOF. Fix d . Let $\Sigma = \{a, b\}$. Suppose $\beta = \omega^d \cdot j_1 + \omega^{d-1} \cdot j_2 + \dots + j_{d+1}$.

Let $L_\beta^1 =$

$$\begin{aligned} & \mathbf{Lang}(a^{3j_1+1}ba^{3j_2+1}b \dots a^{3j_d+1}ba^{3j_{d+1}+2}x) \cup \\ & \quad \mathbf{Lang}(a^{3j_1+3}x) \cup \\ & \quad \mathbf{Lang}(a^{3j_1+1}ba^{3j_2+3}x) \cup \\ \dots & \quad \mathbf{Lang}(a^{3j_1+1}ba^{3j_2+1} \dots a^{3j_d+3}x). \end{aligned}$$

Let $L_\beta^2 =$

$$\begin{aligned} & \mathbf{Lang}(a^{3j_1+1}ba^{3j_2+1}b \dots a^{3j_d+1}ba^{3j_{d+1}+1}x) \cup \\ & \quad \mathbf{Lang}(a^{3j_1+3}x) \cup \\ & \quad \mathbf{Lang}(a^{3j_1+1}ba^{3j_2+3}x) \cup \\ \dots & \quad \mathbf{Lang}(a^{3j_1+1}ba^{3j_2+1} \dots a^{3j_d+3}x). \end{aligned}$$

It is easy to verify that the above L_β^1, L_β^2 satisfy the lemma. ■

We now continue with the proof of the theorem. Fix d and $\alpha \prec \omega^{d+1}$. Let $\mathcal{L} \subseteq \mathbf{TxtEx}_\alpha(\mathbf{M}, \mathbf{F})$. Suppose by way of contradiction that Θ, Ψ witness $PATTERN^{d+1} \leq_{\text{weak}}^{\mathbf{TxtEx}} \mathcal{L}$.

For $\beta \prec \omega^{d+1}$, let L_β^1, L_β^2 be as in Lemma 3.

We now construct a text in stages as described below. Let $\sigma_0 = \Lambda$. Go to Stage 0.

Begin Stage s

1. Suppose $\mathbf{F}(\Theta(\sigma_s)) = \beta$.
2. Search for an extension σ_{s+1} of σ_s such that $\mathbf{F}(\Theta(\sigma_{s+1})) \prec \mathbf{F}(\Theta(\sigma_s))$ and $\text{content}(\sigma_{s+1}) \subseteq L_\beta^2$.
3. If such a σ_{s+1} is found, then go to Stage $s + 1$.

End Stage s

It is easy to verify by induction that $\text{content}(\sigma_s) \subseteq L_{\mathbf{F}(\Theta(\sigma_s))}^1$, and so the search in Step 2 of the construction makes sense. Clearly, there can be only finitely many stages which halt (due to well orderedness of ordinals). Let s be the stage which starts but does not halt. Let T_1 be a text for $L_{\mathbf{F}(\Theta(\sigma_s))}^1$ and T_2 be a text for $L_{\mathbf{F}(\Theta(\sigma_s))}^2$, such that $\sigma_s \subseteq T_1$ and $\sigma_s \subseteq T_2$. Since Θ, Ψ witness the reduction of $PATTERN^{d+1}$ to \mathcal{L} , we must have that $\Theta(T_1)$ and $\Theta(T_2)$ are texts for two distinct languages in \mathcal{L} . Moreover, since Step 2 does not succeed in Stage s , we have $\mathbf{M}(\Theta(T_1)) = \mathbf{M}(\Theta(T_2)) = \mathbf{M}(\Theta(\sigma_s))$. It follows that, $\mathcal{L} \not\subseteq \mathbf{TxtEx}_\alpha(\mathbf{M}, \mathbf{F})$. A contradiction. Thus $PATTERN^{d+1} \not\leq_{\text{weak}}^{\mathbf{TxtEx}} \mathcal{L}$. ■

Now since $LBEFS^d$ can be identified with ordinal mind change complexity of ω^d , we immediately have the following corollary to the above theorem which says that learning

unions of up to $d + 1$ pattern languages is not reducible to the problem of learning *EFS* languages definable by up to d length-bounded clauses.

Corollary 4 *Let $d \in \mathbb{N}^+$. $PATTERN^{d+1} \not\leq_{\text{weak}}^{\text{TxtEx}} LBEFS^d$.*

The above corollary additionally yields the following pleasing corollary that is very natural. It is interesting to note that the following result further establishes intrinsic complexity as a useful measure of the difficulty of learning a class of languages.

Corollary 5 *Let $d \in \mathbb{N}^+$.*

(a) $PATTERN^{d+1} \not\leq_{\text{weak}}^{\text{TxtEx}} PATTERN^d$.

(b) $LBEFS^{d+1} \not\leq_{\text{weak}}^{\text{TxtEx}} LBEFS^d$.

3.3 Topological Properties, Intrinsic Complexity, and Procrastination

In this section, we consider connections between topological properties of learnable classes and their intrinsic complexity. The following notion was introduced by Angluin [Ang80].

Definition 19 \mathcal{L} has *finite thickness* just in case for each $n \in \mathbb{N}$, $\text{card}(\{L \in \mathcal{L} \mid n \in L\})$ is finite.

PATTERN has finite thickness. Angluin [Ang80] showed that if \mathcal{L} is an indexed family of recursive languages and \mathcal{L} has finite thickness then $\mathcal{L} \in \text{TxtEx}$. A more interesting topological notion was introduced by Wright [Wri89] (see also Motoki, Shinohara, and Wright [MSW91]) described below.

Definition 20 [Wri89, MSW91] \mathcal{L} has *infinite elasticity* just in case there exists an infinite sequence of pairwise distinct numbers, $\{w_i \in \mathbb{N} \mid i \in \mathbb{N}\}$, and an infinite sequence of pairwise distinct languages, $\{A_i \in \mathcal{L} \mid i \in \mathbb{N}\}$, such that for each $k \in \mathbb{N}$, $\{w_i \mid i < k\} \subseteq A_k$, but $w_k \notin A_k$. \mathcal{L} is said to have *finite elasticity* just in case \mathcal{L} does not have infinite elasticity.

Wright [Wri89] showed that if a class \mathcal{L} has finite thickness then it has finite elasticity. He further showed that if a class \mathcal{L} is an indexed family of recursive languages and \mathcal{L} has finite elasticity, then $\mathcal{L} \in \text{TxtEx}$.

Now, language classes that are $\leq_{\text{weak}}^{\text{TxtEx}}$ -complete are, in some sense, the most difficult learning problems. Interestingly, we are able to establish that $\leq_{\text{weak}}^{\text{TxtEx}}$ -completeness is also a sufficient condition for infinite elasticity.

Theorem 5 *Suppose \mathcal{L} is $\leq_{\text{weak}}^{\text{TxtEx}}$ -complete. Then \mathcal{L} has infinite elasticity.*

PROOF. Suppose \mathcal{L} is given such that $FIN \leq_{\text{weak}}^{\text{TxtEx}} \mathcal{L}$ as witnessed by Θ and Ψ . Let $X_i = \{x \mid x \leq i\}$.

The following lemma is proved in [JS96b] (Lemma 2(b)).

Lemma 4 *Suppose Θ is an arbitrary enumeration operator, $Y_1 \subseteq Y_2 \subseteq N$ and $Y'_1 \in \Theta(Y_1)$. Then for every finite subset S of Y'_1 , there exists an $Y'_2 \in \Theta(Y_2)$ such that $S \subseteq Y'_2$.*

Thus, we have

$$(\forall i, j \mid j > i)(\forall L \in \Theta(X_i))(\forall \text{finite } S \subseteq L)(\exists L' \in \Theta(X_j))[S \subseteq L'] \quad (1)$$

We consider two cases:

Case 1: There exists an i , there exists an $L \in \Theta(X_i)$, there exists a finite $S \subseteq L$, such that

$$(\forall j > i)(\forall L' \in \Theta(X_j))[S \subseteq L' \Rightarrow L' \subseteq L]$$

In this case let i, L, S be as witnessing above.

Let A_0 be an element of $\Theta(X_{i+1})$ such that $S \subseteq A_0$. Note that such a A_0 exists (by (1) above). Define w_k, A_{k+1} inductively as follows.

Now $A_k \subseteq L$ (by induction and hypothesis of this case) and $A_k \neq L$ (since X_i and X_{i+k+1} are distinct). Let w_k be an element of $L - A_k$. Let A_{k+1} be a member of $\Theta(X_{i+k+2})$ such that $S \cup \{w_0, \dots, w_k\} \subseteq A_{k+1}$. Note that there exists such a A_{k+1} , since $S \cup \{w_0, \dots, w_k\} \subseteq L$, $L \in \Theta(X_i)$ and $X_i \subseteq X_{i+k+2}$ (by (1) above).

It is now easy to observe that A_k, w_k witness that \mathcal{L} is of infinite elasticity.

Case 2: Not Case 1. Thus, for all i , for all $L \in \Theta(X_i)$, for all finite $S \subseteq L$, there exists a $j > i$, there exists a $L' \in \Theta(X_j)$, such that $S \subseteq L'$ and $L' \not\subseteq L$.

Let A_0 be a member of $\Theta(X_0)$. Define w_k and A_{k+1} inductively as follows.

Let L' be as witnessing the Case 2, when $L = A_k$, and $S = \{w_0, \dots, w_{k-1}\}$.

Let $A_{k+1} = L'$ and w_k be a member in $L' - A_k$.

It is now easy to observe that A_k, w_k witness that \mathcal{L} is of infinite elasticity.

From the above cases it follows that \mathcal{L} has infinite elasticity. ■

Classes that have infinite elasticity are not necessarily identifiable. However, it is interesting to ask: Are all identifiable classes that have infinite elasticity also $\leq_{\text{weak}}^{\text{TxtEx}}$ -complete? The following result answers this question negatively.

Theorem 6 *There exist a class \mathcal{L} such that $\mathcal{L} \in \text{TxtEx}$ and \mathcal{L} has infinite elasticity, but \mathcal{L} is not $\leq_{\text{weak}}^{\text{TxtEx}}$ -complete.*

PROOF. (Sketch) For $i \in N$, consider the language:

$$L_i = \{\langle 0, i \rangle\} \cup \{\langle 1, x \rangle \mid x \leq i\}$$

Consider the class $\mathcal{L} = \{L_i \mid i \in N\}$. It is easy to verify that \mathcal{L} has infinite elasticity. Also, \mathcal{L} cannot be $\leq_{\text{weak}}^{\mathbf{TxtEx}}$ -complete since it can be \mathbf{TxtEx} -identified using 0 mind changes. ■

Finite elasticity is a sufficient condition for identification of indexed families of recursive languages. Also, the property of finite elasticity is preserved under finite unions. We have seen that for each $d > 0$, $PATTERN^d \in \mathbf{TxtEx}_{\omega^d}$. It would be interesting to investigate if for each indexed family of recursive languages \mathcal{L} that has finite elasticity, if there is an i such that $\mathcal{L} \in \mathbf{TxtEx}_{\omega^i}$. The answer to this question turns out to be negative as implied by the following result.

Theorem 7 *There exists a class \mathcal{L} such that the following hold:*

- (a) \mathcal{L} is an indexed family of recursive languages;
- (b) \mathcal{L} has finite elasticity; and
- (c) for each $i > 0$, $\mathcal{L} \notin \mathbf{TxtEx}_{\omega^i}$.

PROOF. Let $X_L^i = \{\langle i, x \rangle \mid x \in L\}$. Let $\mathcal{L} = \{X_L^i \mid i > 0 \wedge L \in PATTERN^i\}$. Since, for each i , $PATTERN^i$ has finite elasticity, it follows that \mathcal{L} has finite elasticity. Since $PATTERN^{i+1} \notin \mathbf{TxtEx}_{\omega^i}$, it follows that $\mathcal{L} \notin \mathbf{TxtEx}_{\omega^i}$. ■

4 Conclusion

Interesting connections between three distinct areas of inductive inference, viz., learnability of *EFS* subclasses, intrinsic complexity, and ordinal mind change complexity were established. These results show that the abstract ideas of intrinsic complexity and ordinal mind change complexity can be employed to gain an insight into the learnability of concrete language classes. Another contribution of the paper is the establishment of some connections between the topological properties of learnable classes of indexed families and the notion of intrinsic complexity and ordinal mind change complexity. More results of this nature will enhance our understanding of identification from positive data. Also, since *EFS* subclasses have counterparts in standard logic programming, the results presented in this paper form a basis for employing inductive inference to analyze identification in the limit results for inductive logic programming.

References

- [AMS⁺92] S. Arikawa, S. Miyano, A. Shinohara, T. Shinohara, and A. Yamamoto. Algorithmic learning theory with elementary formal systems. *IEICE Trans. Inf. and Syst.*, E75–D No. 4:405–414, 1992.

- [Ang80] D. Angluin. Finding patterns common to a set of strings. *Journal of Computer and System Sciences*, 21:46–62, 1980.
- [Ari70] S. Arikawa. Elementary formal systems and formal languages—simple formal systems. *Memoirs of the Faculty of Science, Kyushu University Series A*, 24:47–75, 1970.
- [Ari89] H. Arimura. Completeness of depth-bounded resolution in logic programming. In *Proceedings of the 6th Conference, Japan Soc. Software Sci. Tech.*, pages 61–64, 1989.
- [AS94] H. Arimura and T. Shinohara. Inductive inference of Prolog programs with linear data dependency from positive data. In *Proc. Information Modelling and Knowledge Bases V*, pages 365–375. IOS Press, 1994.
- [ASMS91] S. Arikawa, T. Shinohara, S. Miyano, and A. Shinohara. More about learning elementary formal systems. In G. Brewka, K. P. Jantke, and P. H. Schmitt, editors, *Nonmonotonic and Inductive Logic*, pages 107–117. Springer-Verlag, 1991. Lecture Notes in Artificial Intelligence 659.
- [ASY92] S. Arikawa, T. Shinohara, and A. Yamamoto. Learning elementary formal systems. *Theoretical Computer Science*, 95:97–113, 1992.
- [Blu67] M. Blum. A machine-independent theory of the complexity of recursive functions. *Journal of the ACM*, 14:322–336, 1967.
- [CJS95] J. Case, S. Jain, and M. Suraj. Not-so-nearly-minimal-size program inference. In Klaus P. Jantke and Steffen Lange, editors, *Algorithmic Learning for Knowledge-Based Systems*, volume 961 of *Lecture Notes in Artificial Intelligence*, pages 77–96. Springer-Verlag, 1995.
- [FKS95] R. Freivalds, E. Kinber, and C. H. Smith. On the intrinsic complexity of learning. In Paul Vitanyi, editor, *Proceedings of the Second European Conference on Computational Learning Theory*, pages 154–169. Springer-Verlag, March 1995. Lecture Notes in Artificial Intelligence 904.
- [FS92] R. Freivalds and C. Smith. On the role of procrastination for machine learning. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory, Pittsburgh, Pennsylvania*, pages 363–376. ACM Press, July 1992.
- [FS93] R. Freivalds and C. Smith. On the role of procrastination in machine learning. *Information and Computation*, pages 237–271, 1993.

- [Gol67] E. M. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.
- [HU79] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley Publishing Company, 1979.
- [JS94] S. Jain and A. Sharma. On the intrinsic complexity of language identification. In *Proceedings of the Seventh Annual Conference on Computational Learning Theory, New Brunswick, New Jersey*, pages 278–286. ACM-Press, July 1994.
- [JS95] S. Jain and A. Sharma. The structure of intrinsic complexity of learning. In Paul Vitanyi, editor, *Computational Learning Theory, Second European Conference, EuroCOLT’95, Barcelona, Spain*, pages 169–181. Springer-Verlag, March 1995. Lecture Notes in Artificial Intelligence 904.
- [JS96a] S. Jain and A. Sharma. On the intrinsic complexity of language identification. *Journal of Computer and System Sciences*, 52(3):393–402, June 1996. Special issue on Computational Learning Theory, 1994.
- [JS96b] S. Jain and A. Sharma. The structure of intrinsic complexity of learning. *Journal of Symbolic Logic*, 1996. Accepted. Preliminary version in [JS95].
- [Kle38] S. C. Kleene. Notations for ordinal numbers. *Journal of Symbolic Logic*, 3:150–155, 1938.
- [KR96] M. R. K. Krishna Rao. A class of Prolog programs inferable from positive data. In A. Arikawa and A. Sharma, editors, *Proceedings of the Seventh International Workshop on Algorithmic Learning Theory*, pages 272–284. Lecture Notes in Artificial Intelligence, No. 1160, Springer-Verlag, 1996.
- [MSW91] T. Motoki, T. Shinohara, and K. Wright. The correct definition of finite elasticity: Corrigendum to identification of unions. In L. Valiant and M. Warmuth, editors, *Proceedings of the Fourth Annual Workshop on Computational Learning Theory, Santa Cruz, California*, page 375. Morgan Kaufman, 1991.
- [MY78] M. Machtey and P. Young. *An Introduction to the General Theory of Algorithms*. North Holland, New York, 1978.
- [Rog67] H. Rogers. *Theory of Recursive Functions and Effective Computability*. McGraw-Hill, New York, 1967. Reprinted, MIT Press 1987.
- [Sac90] G. E. Sacks. *Higher Recursion Theory*. Springer-Verlag, 1990.
- [Shi86] T. Shinohara. *Studies on Inductive Inference from Positive Data*. PhD thesis, Kyushu University, Kyushu, Japan, 1986.

- [Shi91] T. Shinohara. Inductive inference of monotonic formal systems from positive data. *New Generation Computing*, 8:371–384, 1991.
- [Shi94] T. Shinohara. Rich classes inferable from positive data: Length-bounded elementary formal systems. *Information and Computation*, 108:175–186, 1994.
- [Smu61] R. Smullyan. *Theory of Formal Systems, Annals of Mathematical Studies, No. 47*. Princeton, NJ, 1961.
- [Wri89] K. Wright. Identification of unions of languages drawn from an identifiable class. In R. Rivest, D. Haussler, and M. K. Warmuth, editors, *Proceedings of the Second Annual Workshop on Computational Learning Theory, Santa Cruz, California*, pages 328–333. Morgan Kaufmann Publishers, Inc., 1989.