The Structure of Intrinsic Complexity of Learning

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

Limiting identification of r.e. indexes for r.e. languages (from a presentation of elements of the language) and limiting identification of programs for computable functions (from a graph of the function) have served as models for investigating the boundaries of learnability. Recently, a new approach to the study of "intrinsic" complexity of identification in the limit has been proposed. This approach, instead of dealing with the resource requirements of the learning algorithm, uses the notion of reducibility from recursion theory to compare and to capture the intuitive difficulty of learning various classes of concepts. Freivalds, Kinber, and Smith have studied this approach for function identification and Jain and Sharma have studied it for language identification.

The present paper explores the structure of these reducibilities in the context of language identification. It is shown that there is an infinite hierarchy of language classes that represent learning problems of increasing difficulty. It is also shown that the language classes in this hierarchy are incomparable, under the reductions introduced, to the collection of pattern languages.

Richness of the structure of intrinsic complexity is demonstrated by proving that any finite, acyclic, directed graph can be embedded in the reducibility structure. However, it is also established that this structure is not dense. The question of embedding any infinite, acyclic, directed graph is open.

1 Introduction

Recently, a new approach to the study of "intrinsic" complexity of learning has been proposed. This approach can be traced to the work of Freivalds [Fre91] and was 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, JS96].

Instead of concentrating on the resource requirement complexity of the learning algorithm, this approach uses the notion of reducibility from recursion theory to compare the learning difficulty of the concept classes being learned. Since the treatment of languages turns out to be more general than that of functions (as functions can be treated as single valued total languages), we limit our investigation to languages in the present paper.

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. We present an informal description of these reductions. To facilitate our discussion, we first introduce some technical details about language identification in the limit.

L, with or without decorations, ranges over recursively enumerable languages over the set of natural numbers. An r.e. index for a language L is referred to as a *grammar* (acceptor) for L. Informally, a *text* for a language L is just an infinite sequence, with possible repetitions, of all and only the elements of L. A learning machine is an algorithmic device. Elements of a text are sequentially fed to a learning machine one element at a time. The learning machine, as it receives elements of the text, outputs an infinite sequence of grammars. If the sequence of grammars emitted by the learning machine converges to a single correct grammar for the language whose text is fed to the machine, then the machine is said to *identify* the text. A machine is said to *identify* a language just in case it identifies each text for the language. This is essentially Gold's [Gol67] criterion of identification in the limit.

It is also useful to call an infinite sequence of grammars, $g_0, g_1, g_2, \ldots$, *admissible* for a text T just in case the sequence of grammars converges to a single correct grammar for the language whose text is T.

The reductions are based on the idea that for a collection of languages $\mathcal{L}$ to be reducible to $\mathcal{L}'$, we should be able to transform texts T for languages in $\mathcal{L}$ to texts T' for languages in $\mathcal{L}'$ and further transform admissible sequences for T' into admissible sequences for T. This is achieved with the help of two enumeration operators. Informally, enumeration operators are algorithmic devices that map infinite sequences of objects (for example, texts and infinite sequences of grammars) into infinite sequences of objects. The first operator, Θ, transforms texts for languages in $\mathcal{L}$ into texts for languages in $\mathcal{L}'$. The second operator, Ψ, behaves as follows: if Θ transforms a text T for some language in $\mathcal{L}$ into text T' (for some language in $\mathcal{L}'$), then Ψ transforms admissible sequences for T' into admissible sequences for T.

To see that the above satisfies the intuitive notion of reduction, consider collections $\mathcal{L}$ and $\mathcal{L}'$ such that $\mathcal{L}$ is reducible to $\mathcal{L}'$. We now argue that if $\mathcal{L}'$ is identifiable then so is $\mathcal{L}$. Let learning machine $\mathbf{M}'$ identify $\mathcal{L}'$. Let enumeration operators Θ and Ψ witness the reduction of $\mathcal{L}$ to $\mathcal{L}'$. Then we describe a learning machine $\mathbf{M}$ that **TxtEx**-identifies $\mathcal{L}$. $\mathbf{M}$, upon being fed a text T for some language $L \in \mathcal{L}$, uses Θ to construct a text T' for a language in $\mathcal{L}'$. It then simulates machine $\mathbf{M}'$ on text T' and feeds conjectures of $\mathbf{M}'$ to the operator Ψ to produce its conjectures. It is easy to verify that the properties of Θ, Ψ, and $\mathbf{M}'$ guarantee the success of $\mathbf{M}$ on each text for each language in $\mathcal{L}$.

The above reduction has been 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) $COINIT = \{L \mid (\exists n)[L = \{x \mid x \geq n\}]\}$ (essentially, *COINIT* is the collection of those languages that contain all the natural numbers except a finite initial segment);

(c) *FIN*, the collection of finite languages.

According to the above reduction *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, JS96] that this reduction captures an intuitive sense in which these classes represent learning problems of increasing difficulty. The class *SINGLE* can be identified by a learning machine that can confirm its success. The class *COINIT* cannot be identified by any machine that can confirm its success, but it can be identified by a machine, that after inspecting an element of the language, provides an upper bound on the number of mind changes it will make before converging to a correct grammar. *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 the reduction described above, *FIN* is complete—it poses the most difficult learning problem. It was also shown that the class *COINIT* was equivalent to *PATTERN*, the class of pattern languages introduced by Angluin [Ang80a]. Pattern languages are a useful class of languages and will be described later in the paper.

In the present paper we investigate the structure of the above reductions. We present a series of results that provide sufficient conditions for when a collection of languages is not reducible to another. Using these results we show that there is an infinite chain of language classes that represent learning problems of increasing difficulty. We give an informal description of this chain of languages.

For $i \geq 1$, let FIN_i denote the collection of languages with cardinality less than or equal to i. Then for each $i > 1$, FIN_i is reducible to FIN_{i+1} but FIN_{i+1} is not reducible to FIN_i. This means that FIN_{i+1} is a strictly more difficult learning problem than FIN_i. We also provide an insight into the classes FIN_i and identification in the limit with bounded number of mind changes. If a collection of languages $\mathcal{L}$ can be identified in the limit with no more than n mind changes then $\mathcal{L}$ is reducible to FIN_{n+1}. That is, we can use an algorithm that learns FIN_{n+1} to design an algorithm that learns $\mathcal{L}$.

It is interesting to investigate the relationship between identification in the limit with bounded number of mind changes and identification in the limit where a learning machine can provide an upper bound on the number of mind changes after inspecting an element of the language being learned. In other words, we would like to find out how FIN_i, $i > 1$, and *PATTERN* (or, *COINIT*) compare with respect to the above reduction. It turns out that *PATTERN* is incomparable to FIN_i, for each $i > 1$, with respect to the above reduction. In other words, for each $i > 1$, FIN_i is not reducible to *PATTERN* and *PATTERN* is not reducible to FIN_i.

We demonstrate the richness of this reducibility structure by showing that any finite directed acyclic graph can be embedded in the reducibility structure. However, we also show that the reducibility structure is not *dense* as there are language classes between which there is nothing. More specifically, we show that there exist language classes $\mathcal{L}$ and $\mathcal{L}'$ such that

- $\mathcal{L}$ is reducible to $\mathcal{L}'$ and $\mathcal{L}'$ is not reducible to $\mathcal{L}$, and
- for all $\mathcal{L}''$ such that $\mathcal{L}$ is reducible to $\mathcal{L}''$ and $\mathcal{L}''$ is reducible to $\mathcal{L}'$, either $\mathcal{L}''$ is equivalent to $\mathcal{L}$ or $\mathcal{L}''$ is equivalent to $\mathcal{L}'$.

We also consider a finer notion of reduction called strong reduction in which texts for the same language are mapped to texts for a unique language. Under this stronger version of reduction, it turns out that *FIN* is not complete. We demonstrate an interesting class of languages that is trivially learnable (with 0 mind changes) but is not strong-reducible to *FIN*.

We also note that the question of embedding any infinite, acyclic, directed graph in the reducibility structure is open.

We now proceed formally.

2 Notation and Preliminaries

Any unexplained recursion theoretic notation is from [Rog67]. The symbol N denotes the set of natural numbers, $\{0, 1, 2, 3, \ldots\}$. Unless otherwise specified, e, g, i, j, k, l, m, n, q, r, s, t, w, x, y, with or without decorations[1], range over N. Unless otherwise specified all conventions in this paper regarding range of variables apply for variables with or without decorations. Symbols $\emptyset$, $\subseteq$, $\subset$, $\supseteq$, and $\supset$ denote empty set, subset, proper subset, superset, and proper superset, respectively. Symbols A, L and S range over sets of numbers.

Cardinality of a set S is denoted by $\text{card}(S)$. The maximum and minimum of a set are denoted by $\max(\cdot), \min(\cdot)$, respectively, where $\max(\emptyset) = 0$ and $\min(\emptyset) = \infty$. $\langle \cdot, \cdot \rangle$ denotes an arbitrary, computable, 1–1, onto pairing function from $N \times N$ onto N. Similarly, one can define $\langle \cdot, \ldots, \cdot \rangle$ for encoding multiple tuples of natural numbers onto N. Unless otherwise specified, letters f, F and h range over *total* functions with arguments and values from 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. The letter, p, in some contexts ranges over programs; in other contexts p ranges over total functions with its range being construed as programs. 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 $\{x \leq s \mid \Phi_i(x) < s\}$. $\downarrow$ denotes defined. $\uparrow$ denotes undefined.

We now present concepts from language learning theory. The definition below introduces the concept of a *sequence* of data.

Definition 1

(a) A *sequence* σ is a mapping from an initial segment of N into $(N \cup \{\#\})$. Empty sequence is denoted by Λ.

(b) The *content* of a sequence σ, denoted $\text{content}(\sigma)$, is the set of natural numbers in the range of σ.

(c) The *length* of σ, denoted by $|\sigma|$, is the number of elements in σ. So, $|\Lambda| = 0$.

(d) For $n \leq |\sigma|$, the initial sequence of σ of length n is denoted by $\sigma[n]$. So, $\sigma[0]$ is Λ.

Intuitively, #'s represent pauses in the presentation of data. We let σ, τ, and γ range over finite sequences. SEQ denotes the set of all finite sequences.

Definition 2 A *language learning machine* is an algorithmic device which computes a mapping from SEQ into N.

We let $\mathbf{M}$ range over learning machines.

Definition 3 A *text* T for a language L is a mapping from N into $(N \cup \{\#\})$ such that L is the set of natural numbers in the range of T. The *content* of a text T, denoted $\text{content}(T)$, is the set of natural numbers in the range of T. $T[n]$ denotes the finite initial sequence of T with length n.

[1]Decorations are subscripts, superscripts and the like.

$\mathbf{M}(T[n])$ is interpreted as the grammar (index for an accepting program) conjectured by learning machine $\mathbf{M}$ on initial sequence $T[n]$.

We also need the notion of an infinite sequence of grammars. We let G range over infinite sequences of grammars. Clearly infinite sequences of grammars are essentially infinite sequences over N. Hence, we adopt the machinery defined for sequences and texts over to finite sequences of grammars and infinite sequences of grammars. So, if $G = g_0, g_1, g_2, g_3, \ldots$, then $G[3]$ denotes the sequence g_0, g_1, g_2, $G(3)$ is g_3.

Below we define identification in the limit introduced by Gold [Gol67].

Definition 4 [Gol67]

(a) $\mathbf{M}$ **TxtEx**-identifies a text T just in case $(\exists i \mid W_i = \text{content}(T))\ (\overset{\infty}{\forall} n)[\mathbf{M}(T[n]) = i]$.

(b) $\mathbf{M}$ **TxtEx**-identifies L (written: $L \in \mathbf{TxtEx}(\mathbf{M})$) just in case $\mathbf{M}$ **TxtEx**-identifies each text for L.

(c) $\mathbf{TxtEx} = \{\mathcal{L} \subseteq \mathcal{E} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{TxtEx}(\mathbf{M})]\}$.

Definition 5 *We say that an infinite sequence of grammars $g_0, g_1, \ldots$ is* **TxtEx**-admissible *for text T just in case*

$$(\exists n)[W_{g_n} = \text{content}(T)\ \wedge\ (\forall n' \geq n)[g_{n'} = g_n]].$$

3 Reductions and Previous Results

We first present some technical machinery.

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

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

(i) for all $\sigma, \tau \in$ SEQ, if $\sigma \subseteq \tau$, then $\Theta(\sigma) \subseteq \Theta(\tau)$;

(ii) For all texts T, $\lim_{n\to\infty} |\Theta(T[n])| = \infty$.

By extension, we think of Θ as also defining a mapping from **TEXTS** into **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 now introduce our first reduction.

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 $\mathcal{L} \leq^{\mathbf{TxtEx}}_{\text{weak}} \mathcal{L}'$ just in case there exist operators Θ and Ψ such that for all $T \in \mathcal{T}$ and for all infinite sequence of grammars $G = g_0, g_1, \ldots$, the following two conditions hold:

(a) $\Theta(T) \in \mathcal{T}'$ and

(b) if G is an **TxtEx**-admissible sequence for $\Theta(T)$, then $\Psi(G)$ is an **TxtEx**-admissible sequence for T.

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

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 $\leq^{\mathbf{TxtEx}}_{\text{weak}}$-*hard* iff for all $\mathcal{L}' \in \mathbf{TxtEx}$, $\mathcal{L}' \leq^{\mathbf{TxtEx}}_{\text{weak}} \mathcal{L}$.

(b) We say that $\mathcal{L}$ is $\leq^{\mathbf{TxtEx}}_{\text{weak}}$-*complete* iff $\mathcal{L}$ is $\leq^{\mathbf{TxtEx}}_{\text{weak}}$-hard and $\mathcal{L} \in \mathbf{TxtEx}$.

It should be noted that there is no requirement that Θ map every text for a language in $\mathcal{L}_1$ into texts for a unique language in $\mathcal{L}_2$. 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 $\mathcal{L} \leq^{\mathbf{TxtEx}}_{\text{strong}} \mathcal{L}'$ just in case there exist operators Θ, Ψ such that

(a) Θ and Ψ witness that $\mathcal{L} \leq^{\mathbf{TxtEx}}_{\text{weak}} \mathcal{L}'$, 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 ($\forall$ texts T for L)[$\Theta(T)$ is a text for L'].

In the above case, we also say that Θ and Ψ witness $\mathcal{L} \leq^{\mathbf{TxtEx}}_{\text{strong}} \mathcal{L}'$. We say that $\mathcal{L} \equiv^{\mathbf{TxtEx}}_{\text{strong}} \mathcal{L}'$ iff $\mathcal{L} \leq^{\mathbf{TxtEx}}_{\text{strong}} \mathcal{L}'$ and $\mathcal{L}' \leq^{\mathbf{TxtEx}}_{\text{strong}} \mathcal{L}$.

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

Since **TxtEx** is the only learning paradigm considered in the present paper, we refer to $\leq^{\mathbf{TxtEx}}_{\text{weak}}$ and $\leq^{\mathbf{TxtEx}}_{\text{strong}}$ by $\leq_{\text{weak}}$ and $\leq_{\text{strong}}$, repectively in the sequel. The following theorem summarizes the relevant results about the above reductions.

Theorem 1 [JS94, JS96]

(a) SINGLE $\leq_{\text{strong}}$ *COINIT and COINIT* $\not\leq_{\text{weak}}$ *SINGLE.*

(b) COINIT $\leq_{\text{weak}}$ *FIN and FIN* $\not\leq_{\text{weak}}$ *COINIT.*

(c) FIN is $\leq_{\text{weak}}$*-complete, but not* $\leq_{\text{strong}}$*-complete.*

The reader is referred to [JS94, JS96] for additional examples and for a collection of languages that is complete with respect to strong reduction.

4 Results

To begin with, we present a series of lemmas that give sufficient conditions for when a collection of languages is not reducible to another collection of languages. First, the following simple lemma:

Lemma 1 *Suppose* $\mathcal{L} \subseteq \mathcal{E}$, $\mathcal{L}' \subseteq \mathcal{E}$ *and* $\mathcal{L} \leq_{\text{weak}} \mathcal{L}'$ *as witnessed by* Θ *and* Ψ*. Then*

(a) $(\forall L \subseteq N)(\forall \sigma \mid \text{content}(\sigma) \subseteq L)(\exists L' \in \Theta(L))[\text{content}(\Theta(\sigma)) \subseteq L']$;

(b) $(\forall L \in \mathcal{L})[\Theta(L) \subseteq \mathcal{L}']$;

(c) $(\forall L, L' \in \mathcal{L})[L \neq L' \Rightarrow \Theta(L) \cap \Theta(L') = \emptyset]$.

PROOF. Part (a) follows from definition of $\Theta(L)$. Part (b) follows from the definition of $\leq_{\text{weak}}$.

For part (c) suppose by way of contradiction that $L, L' \in \mathcal{L}$, $L \neq L'$ and $\Theta(L) \cap \Theta(L') \neq \emptyset$. Let T be a text for L and T' be a text for L' such that $\text{content}(\Theta(T)) = \text{content}(\Theta(T'))$. Let G be an admissible sequence of grammars for $\Theta(T)$. Therefore G is also an admissible sequence for $\Theta(T')$. Thus, since Θ, Ψ witness $\mathcal{L} \leq_{\text{weak}} \mathcal{L}'$, $\Psi(G)$ must be admissible sequence for both T and T'. But this is impossible, since $\text{content}(T) = L \neq L' = \text{content}(T')$. ▮

Lemma 2 *Suppose Θ is an enumeration operator.*

(a) Suppose $L \subseteq N$ and $L' \in \Theta(L)$. Then $(\forall \text{ finite } S \subseteq L')(\exists \sigma \mid \text{content}(\sigma) \subseteq L)[S \subseteq \text{content}(\Theta(\sigma)) \subseteq L']$.

(b) Suppose $L_1 \subseteq L_2 \subseteq N$, and $L'_1 \in \Theta(L_1)$. Then for every finite subset S of L'_1, there exists an $L'_2 \in \Theta(L_2)$ such that $S \subseteq L'_2$.

(c) Suppose $L_1 \subseteq L_2 \subseteq N$. Suppose further that $\Theta(L_1)$ consists only of finite languages. Then for all $L'_1 \in \Theta(L_1)$, there exists an $L'_2 \in \Theta(L_2)$ such that $L'_1 \subseteq L'_2$.

PROOF. (a) Assume the hypothesis. Thus there exists a text T for L such that $\text{content}(\Theta(T)) = L'$. Let n be such that $\text{content}(\Theta(T[n])) \supseteq S$ (there exists such an n, since some finite initial segment of $\Theta(T)$ contains all the elements of S). Taking $T[n]$ as σ satisfies part (a).

(b) Assume the hypothesis. Consider any finite subset S of L'_1. By part (a) there exists a σ such that $\text{content}(\sigma) \subseteq L_1$, and $\text{content}(\Theta(\sigma)) \supseteq S$. Consider a text T_2 for L_2 such that T_2 is an extension of σ. Note that there exists such a T_2 since, $\text{content}(\sigma) \subseteq L_1 \subseteq L_2$. Now $\text{content}(\Theta(T_2)) \supseteq \text{content}(\Theta(\sigma)) \supseteq S$. Since $\text{content}(\Theta(T_2)) \in \Theta(L_2)$, part (b) follows.

(c) follows from part (b). ▮

Corollary 1 *Let $\mathcal{L} \subseteq \mathcal{E}, \mathcal{L}' \subseteq \mathcal{E}$.*

(a) Suppose $\mathcal{L} \leq_{\text{strong}} \mathcal{L}'$ as witnessed by Θ and Ψ. Suppose $L_1, L_2 \in \mathcal{L}$ and $L_1 \subseteq L_2$. Let S_1, S_2 be such that $\Theta(L_1) = \{S_1\}$ and $\Theta(L_2) = \{S_2\}$. Then $S_1 \subseteq S_2$.

(b) Suppose $\mathcal{L} \leq_{\text{weak}} \mathcal{L}'$ as witnessed by Θ and Ψ. Suppose $L_1, L_2 \in \mathcal{L}$ and $L_1 \subseteq L_2$. Further suppose that $\mathcal{L}'$ consists only of finite languages. Then, for every $S_1 \in \Theta(L_1)$, there exists an $S_2 \in \Theta(L_2)$ such that $S_1 \subseteq S_2$.

PROOF. Part (a) follows using Lemma 2(b). Part (b) follows using Lemma 2(c). ▮

We next introduce a technical definition about a structural property of collections of languages:

Definition 10 A *chain* is a sequence of languages $L_1, L_2, \ldots, L_j$, such that $L_1 \subset L_2 \subset \cdots \subset L_j$.

If $L_1, L_2, \ldots, L_j$ form a chain, then we also refer to them as a j-chain.

We say that two chains $L_1, L_2, \ldots, L_j$ and $L'_1, L'_2, \ldots, L'_k$ are independent iff they do not contain any language in common.

We say that $\mathcal{L}$ contains a j-chain, iff it contains languages $L_1, L_2, \ldots, L_j$ which form a j-chain. Similarly, we say that $\mathcal{L}$ contains k-independent j-chains iff, for $1 \leq r \leq k$, $1 \leq i \leq j$, $\mathcal{L}$ contains languages L_i^r, such that, for $1 \leq r \leq k$, $L_1^r, L_2^r, \ldots, L_j^r$ form pairwise-independent chains.

The next lemma gives sufficient condition for non-reducibility in the strong sense. It says for $\mathcal{L} \leq_{\text{strong}} \mathcal{L}'$, $\mathcal{L}'$ must contain at least as many pairwise-independent j-chains, as $\mathcal{L}$.

Lemma 3 *Let $j > 0$. Suppose $\mathcal{L}$ contains k pairwise-independent j-chains and $\mathcal{L} \leq_{\text{strong}} \mathcal{L}'$. Then $\mathcal{L}'$ also has k pairwise-independent j-chains.*

PROOF. Suppose Θ and Ψ witness that $\mathcal{L} \leq_{\text{strong}} \mathcal{L}'$. Suppose $\mathcal{L}$ contains k pairwise-independent j-chains.

For, $1 \leq r \leq k$, $1 \leq i \leq j$, let L_i^r be distinct languages in $\mathcal{L}$ such that

$$(\forall r \mid 1 \leq r \leq k)(\forall i \mid 1 \leq i < j)[L_i^r \subset L_{i+1}^r]$$

Now, $\Theta(L_i^r)$ (which is a subset of $\mathcal{L}'$) contains exactly one language. Let S_i^r denote the language in $\Theta(L_i^r)$. By Lemma 1(c) we have that S_i^r are pairwise distinct. Moreover, by Corollary 1(a) we have that

$$(\forall r \mid 1 \leq r \leq k)(\forall i \mid 1 \leq i < j)[S_i^r \subseteq S_{i+1}^r]$$

It follows that $\mathcal{L}'$ contains k pairwise-independent j-chains. ∎

A slightly weaker version of the above lemma holds for weak reduction.

Lemma 4 *Suppose $\mathcal{L}, \mathcal{L}' \subseteq \mathcal{E}$. Suppose $\mathcal{L} \leq_{\text{weak}} \mathcal{L}'$. Suppose further that $\mathcal{L}$ contains k pairwise-independent j-chains, and $\mathcal{L}'$ consists only of finite languages. Then, $\mathcal{L}'$ contains k pairwise-independent j-chains.*

PROOF. This lemma can be proved along the lines of the proof of Lemma 3, except that this time we use Corollary 1(b). We omit the details. ∎

Using Lemma 4 we now show that there exists an infinite sequence of language classes, $\mathcal{L}_1, \mathcal{L}_2, \ldots$, such that, for each $i \geq 1$, $\mathcal{L}_i$ is strong-reducible to $\mathcal{L}_{i+1}$ but $\mathcal{L}_{i+1}$ is not even weak-reducible to $\mathcal{L}_i$. Consider the following definition:

Definition 11 *For $i > 1$, let $FIN_i = \{L \mid \text{card}(L) \leq i\}$.*

Theorem 2 *For each $i \geq 1$, $FIN_i \leq_{\text{strong}} FIN_{i+1}$ and $FIN_{i+1} \not\leq_{\text{weak}} FIN_i$.*

PROOF. Since $FIN_i \subseteq FIN_{i+1}$ we have $FIN_i \leq_{\text{strong}} FIN_{i+1}$. Now, FIN_{i+1} contains an $(i+2)$-chain whereas FIN_i does not. The theorem follows using Lemma 4. ∎

The above result describes an infinite sequence of language classes that represent learning problems of increasing difficulty. It can be shown that if a collection of languages, $\mathcal{L}$, can be identified with no more than n mind changes, then $\mathcal{L} \leq_{\text{weak}} FIN_{n+1}$.

We next briefly consider the class, *PATTERN*, of pattern languages introduced by Angluin [Ang80a]. Suppose V is a countably infinite set of variables and C is a nonempty *finite* set of constants, such that $V \cap C = \emptyset$. *Notation*: For a set X over variables and constants, X^* denotes the set of strings over X, and X^+ denotes the set of non-empty strings over X. Any $w \in (V \cup C)^+$ is called a *pattern*. Suppose f is a mapping from $(V \cup C)^+$ to C^+, such that, for all $a \in C$, $f(a) = a$ and, for each $w_1, w_2 \in (V \cup C)^+$, $f(w_1 \cdot w_2) = f(w_1) \cdot f(w_2)$, where $\cdot$ denotes concatenation of strings. Let PatMap denote the collection of all such mappings f.

Let code denote a 1-1 onto mapping from strings in C^* to N. The language associated with the pattern w is defined as $L(w) = \{\text{code}(f(w)) \mid f \in \text{PatMap}\}$. Then, $PATTERN = \{L(w) \mid w \text{ is a pattern}\}$.

Angluin [Ang80b] showed that $PATTERN \in$ **TxtEx**. It was shown in [JS94, JS96] that $COINIT \equiv_{\text{strong}} PATTERN$. This result yielded the insight that pattern languages and co-initial languages have similar structure so far as learnability is concerned. Indeed, a machine learning the class *PATTERN* can be modified to calculate an upper bound on the number of mind changes after examining the first element of the language being identified. This is because the pattern that generates the language being identified is of length less than or equal to the length of the element examined. This knowledge can be translated into an upper bound on the number of mind changes that the learner may make before converging to the correct pattern.

So, from Theorem 1, it is clear that *SINGLE* is strong-reducible to *PATTERN* but *PATTERN* is not even weak-reducible to *SINGLE*, thereby implying that *PATTERN* is a strictly more difficult learning task than *SINGLE*. It is interesting to investigate where *PATTERN* lies with respect to FIN_i, $i > 1$. The next theorem shows that for $i > 1$, *PATTERN* is incomparable to FIN_i with respect to weak-reduction.

Theorem 3 *Let $i > 1$. Then,*

(a) $PATTERN \not\leq_{\text{weak}} FIN_i$;

(b) $FIN_i \not\leq_{\text{weak}} PATTERN$.

Proof. The class of pattern languages has an j-chain for each $j > 1$ (consider the patterns ax, aax, $aaax$, ..., where a is a constant and x is a variable). However, the class FIN_i does not contain an $(i+2)$-chain. It follows using Lemma 4 that *PATTERN* is not weak-reducible to FIN_i.

We next show that FIN_2 is not weak-reducible to *PATTERN*. Suppose by way of contradiction that Θ and Ψ witness $FIN_2 \leq_{\text{weak}} PATTERN$. Let σ be such that $\text{content}(\sigma) = \{1\}$, and $\text{content}(\Theta(\sigma)) \neq \emptyset$. There exists such a σ, since $\{1\} \in FIN_2$, and $\emptyset \notin PATTERN$.

Now consider any language $L \in FIN_2$ such that $\{1\} \subseteq L$. By Lemma 2(b), we have that $\text{content}(\Theta(\sigma)) \subseteq L'$, for some $L' \in \Theta(L)$. Moreover, these L' must be distinct for distinct L (Lemma 1(c)). Thus, since there are infinitely many languages in FIN_2 which have $\{1\}$ as a subset, we must have infinitely many pattern languages which have $\text{content}(\Theta(\sigma))$ as a subset. But this is not true. It follows that Θ and Ψ cannot witness $FIN_2 \leq_{\text{weak}} PATTERN$. ∎

A slightly complicated modification of idea used in the above proof can be used to show that there are language classes that are between *SINGLE* and FIN_2 but are incomparable to *PATTERN*.

One can view the reducibility structure as a directed graph, where nodes represent language classes, and an edge from $\mathcal{L}$ to $\mathcal{L}'$ denotes the fact that $\mathcal{L}$ is (weak, strong) reducible to $\mathcal{L}'$.

Theorem 4 shows that the structure of intrinsic complexity is very rich as any finite acyclic directed graph can be embedded in this structure. Theorem 4 uses the following lemma.

Lemma 5 *Suppose $n > 1$ is given. Then there exist language classes $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_n$, such that*

$$(\forall i \mid 1 \leq i \leq n)[\mathcal{L}_i \not\leq_{\text{weak}} \bigcup_{1 \leq j \leq n, j \neq i} \mathcal{L}_j].$$

Proof.

Let $L_{\langle j,i,l \rangle} = \{\langle j, i, x \rangle \mid x \leq l\}$.

Let $\mathcal{S}_{\langle j,i \rangle} = \{L_{\langle j,i,l \rangle} \mid l < j\}$.

Note the following properties, (A) – (E), of $\mathcal{S}_{\langle j,i \rangle}$.

(A) Languages in $\mathcal{S}_{\langle j,i \rangle}$ form a j-chain;

(B) If $L \in \mathcal{S}_{\langle j,i \rangle}$ and $L' \in \mathcal{S}_{\langle j',i' \rangle}$, then $L \cap L' \neq \emptyset \iff (j = j' \ \wedge \ i = i')$;

(C) $\mathcal{S}_{\langle k,i \rangle}$, for $k < j$ does not contain a j-chain.

(D) For $k > j$, $\mathcal{S}_{\langle k,i \rangle}$, does not contain $1 + \lfloor \frac{k}{j} \rfloor$ pairwise-independent j-chains.

(E) All languages in $\mathcal{S}_{\langle j,i \rangle}$ are finite.

Now define $\mathcal{L}_j = \bigcup_{i<n^{3(n+1-j)}} \mathcal{S}_{\langle j,i \rangle}$.

Due to property (B) above, every nonempty chain in $\bigcup_{1\leq j \leq n} \mathcal{L}_j$ is contained in some $\mathcal{L}_j$. Now we have:

(F) Number of j-chains in $\mathcal{L}_j$ is $n^{3(n+1-j)}$.

(G) Number of j-chains in $\bigcup_{1\leq i\leq n, i\neq j} \mathcal{L}_i$ is

$$\sum_{j<k\leq n} \lfloor \frac{k}{j} \rfloor * n^{3(n+1-k)} \leq n * n * n^{3(n-j)} < n^{3(n+1-j)}$$

Lemma now follows using Lemma 4. ∎

Theorem 4 *Every finite acyclic directed graph H can be embedded in the reducibility structure.*

Proof. Without loss of generality assume that H is transitive (otherwise just take the transitive closure of H). Let n denote the number of nodes in H. Let $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_n$ be languages classes such that,

$$(\forall i \mid 1 \leq i \leq n)[\mathcal{L}_i \not\leq_{\text{weak}} \bigcup_{1\leq j\leq n, j\neq i} \mathcal{L}_j].$$

There exist such $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_n$, by Lemma 5.

Let the nodes of the directed graph H be labeled 1 to n. Let E denote the edges of H. We now define the classes $\mathcal{S}_j$, $1 \leq j \leq n$.

Define $\mathcal{S}_j = \mathcal{L}_j \cup \bigcup_{(i,j)\in E} \mathcal{L}_i$. It is easy to see from the property of $\mathcal{L}_j$'s that

$$[(i,j) \in E \Rightarrow \mathcal{S}_i \leq_{\text{strong}} \mathcal{S}_j]$$

and

$$[(i,j) \notin E \Rightarrow \mathcal{S}_i \not\leq_{\text{weak}} \mathcal{S}_j]$$

Theorem follows. ∎

Although the above theorem shows that the intrinsic complexity of language identification is rich, the next result establishes that this structure is not *dense*, that is, there exist two language classes, $\mathcal{L}$ and $\mathcal{L}'$, that satisfy the following properties:

(a) $\mathcal{L}$ is strong-reducible to $\mathcal{L}'$ but $\mathcal{L}'$ is not even weak-reducible to $\mathcal{L}$.

(b) There is no language class between $\mathcal{L}$ and $\mathcal{L}'$ with respect to either strong or weak reduction.

We give the proof only for strong reduction. Straightforward modification of the proof works for weak reduction.

Theorem 5 *For $i > 0$, let $L_i = \{i\}$. Let $L_0 = \{1, 0\}$. Let $\mathcal{L} = \{L_i \mid i > 0\}$. Let $\mathcal{L}' = \{L_0\} \cup \mathcal{L}$. (Note that $\mathcal{L} \leq_{\text{strong}} \mathcal{L}'$, but $\mathcal{L}' \not\leq_{\text{weak}} \mathcal{L}$). Then for all $\mathcal{S}$ such that $\mathcal{L} \leq_{\text{strong}} \mathcal{S} \leq_{\text{strong}} \mathcal{L}'$, either $\mathcal{S} \equiv_{\text{strong}} \mathcal{L}$ or $\mathcal{S} \equiv_{\text{strong}} \mathcal{L}'$.*

Proof. Clearly, $\mathcal{L} \leq_{\text{strong}} \mathcal{L}'$ and by Lemma 4, $\mathcal{L}' \not\leq_{\text{weak}} \mathcal{L}$. Suppose $\mathcal{S}$ is such that $\mathcal{L} \leq_{\text{strong}} \mathcal{S} \leq_{\text{strong}} \mathcal{L}'$. We now show that either $\mathcal{S} \equiv_{\text{strong}} \mathcal{L}$ or $\mathcal{S} \equiv_{\text{strong}} \mathcal{L}'$

Suppose Θ and Ψ witness that $\mathcal{L} \leq_{\text{strong}} \mathcal{S}$ and Θ' and Ψ' witness that $\mathcal{S} \leq_{\text{strong}} \mathcal{L}'$.

Since $\mathcal{S} \leq_{\text{strong}} \mathcal{L}'$, by Lemma 3, we have that $\mathcal{S}$ contains no 3-chain and contains at most one 2-chain. We consider two cases.

Case 1: There is exactly one 2-chain in $\mathcal{S}$.

In this case we show that $\mathcal{L}' \leq_{\text{strong}} \mathcal{S}$. Let $S_1 \subset S_0$ be the 2-chain in $\mathcal{S}$. Clearly, by Lemma 3, we must have $\Theta'(S_0) = \{L_0\}$ and $\Theta'(S_1) = \{L_1\}$. Since L_0, L_1 are finite, there exist σ_0, σ_1 such that $\text{content}(\sigma_0) \subseteq S_0$, $\text{content}(\sigma_1) \subseteq S_1$, $\text{content}(\Theta'(\sigma_0)) = L_0$, and $\text{content}(\Theta'(\sigma_1)) = L_1$. Let $S_0' = \text{content}(\sigma_0)$ and $S_1' = \text{content}(\sigma_1)$.

Claim 1 *(a) For all $L \in \mathcal{S} - \{S_0\}$, $S_0' \not\subseteq L$.*
(b) For all $L \in \mathcal{S} - \{S_0, S_1\}$, $S_1' \not\subseteq L$.

Proof. We show part (a). Part (b) can be shown similarly.

Suppose by way of contradiction, $S_0' \subseteq L$, where $L \in \mathcal{S} - \{S_0\}$. Then, there exists a text T for L such that $\sigma_0 \subset T$. But then we have $L_0 \subseteq \text{content}(\Theta'(T))$. This, by definition of $\mathcal{L}'$ implies $L_0 = \text{content}(\Theta'(T)) \in \Theta(L)$. But then $\Theta(S_0) \cap \Theta(L) \neq \emptyset$, a contradiction to Lemma 1. □

Let $m = \max(\{i \mid i \geq 1 \ \wedge\ [S_0 \in \Theta(L_i) \ \vee\ S_1 \in \Theta(L_i)]\})$. Note that m is well defined since S_0 or S_1 can belong to at most one $\Theta(L_i)$ (Lemma 1).

We now describe two operators Θ'' and Ψ'' that witness the strong-reduction of $\mathcal{L}'$ to $\mathcal{S}$. Θ'' and Ψ'' will be suitable modifications of Θ and Ψ to accommodate L_0, L_1. Let Θ'' be such that the following is satisfied. Note that it is easy to construct such a Θ''.

(a) $\Theta''(L_0) = S_0$;

(b) $\Theta''(L_1) = S_1$;

(c) For $k > 1$, $\Theta''(L_k) = \Theta(L_{k+m})$.

We now construct Ψ''. Let f be a recursive function such that $f(i)$ is a grammar for L_i. Suppose $G = g_0, g_1, g_2, \ldots$. Suppose $\Psi(G) = g_0', g_1', g_2', \ldots$. We define $\Psi''(G) = g_0'', g_1'', g_2'', \ldots$ as follows:

$$g_n'' = \begin{cases} f(0), & \text{if } S_0' \subseteq W_{g_n}^n; \\ f(1), & \text{if } S_0' \not\subseteq W_{g_n}^n \text{ and } S_1' \subseteq W_{g_n}^n; \\ f(k), & \text{if } S_0' \not\subseteq W_{g_n}^n \text{ and } S_1' \not\subseteq W_{g_n}^n \text{ and } k+m = \min(W_{g_n'}^n). \end{cases}$$

We now claim that Θ'', Ψ'' witness that $\mathcal{L}' \leq_{\text{strong}} \mathcal{S}$. Clearly, for all i, $\Theta''(L_i)$ contains exactly one language. Moreover, $\Theta''(L_i)$ are pairwise disjoint (since, $\Theta(L_i)$ are pairwise disjoint, and we know that S_0, S_1 do not belong to $\Theta(L_{k+m})$, for $k > 1$, by definition of m).

Now if G is admissible sequence for $\Theta''(L_0) = S_0$, then clearly, for large enough n, g_n'' as defined above is $f(0)$.

If G is admissible sequence for $\Theta''(L_1) = S_1$, then clearly, for large enough n, g_n'' as defined above is $f(1)$.

If G is admissible sequence for $\Theta''(L_k) = \Theta(L_{m+k})$, $k > 1$, then $\Psi(G)$ is admissible sequence for L_{m+k}, and thus, for large enough n, g_n'' as defined above is $f(k)$.

It follows that Θ'' and Ψ'' witness that $\mathcal{L}' \leq_{\text{strong}} \mathcal{S}$.

Case 2: There is no 2-chain in $\mathcal{L}''$.

In this case we show that $\mathcal{S} \leq_{\text{strong}} \mathcal{L}$. Assume without loss of generality that $L_0 \in \Theta'(S_0)$, for some $S_0 \in \mathcal{S}$ (otherwise, Θ', Ψ' witness that $\mathcal{S} \leq_{\text{strong}} \mathcal{L}$). We now consider two subcases.
Subcase 2.1: For all $S_1 \in \mathcal{S}$, $L_1 \notin \Theta'(S_1)$.

The idea for this subcase is to treat L_0 as L_1 and grammars for L_1 as grammars for L_0. Let Θ''' be defined so that, for all σ, content($\Theta'''(\sigma)$) = content($\Theta'(\sigma)$) $- \{0\}$.

Let f be a recursive function such that

$$W_{f(g)} = \begin{cases} W_g, & \text{if } 1 \notin W_g; \\ W_g \cup \{0\}, & \text{if } 1 \in W_g. \end{cases}$$

Now define Ψ''' as follows: $\Psi'''(g_0, g_1, g_2, \ldots) = \Psi'(f(g_0), f(g_1), f(g_2), \ldots)$. It is easy to verify that Θ''', Ψ''' witness that $\mathcal{S} \leq_{\text{strong}} \mathcal{L}$.

Subcase 2.2: For some $S_1 \in \mathcal{S}$, $L_1 \in \Theta'(S_1)$.

Let σ_0 be such that content(σ_0) $\subseteq S_0$ and content($\Theta'(\sigma_0)$) $= L_0$. Let σ_1 be such that content(σ_1) $\subseteq S_1$ and content($\Theta(\sigma_1)$) $= L_1$. Note that there exist such σ_0 and σ_1. Let S'_0 = content(σ_0). Let $S'_1 \subseteq S_1$ be a finite superset of content(σ_1) such that $S'_1 \not\subseteq S_0$. Note that there exists such a S'_1, since $S_1 \not\subseteq S_0$. Moreover, we have,

$$(\forall L \in \mathcal{S} - \{S_0\})[S'_0 \not\subseteq L] \text{ and } (\forall L \in \mathcal{S} - \{S_1\})[S'_1 \not\subseteq L]$$

We describe Θ''' and Ψ''' that witness $\mathcal{S} \leq_{\text{strong}} \mathcal{L}$. Θ''' and Ψ''' will be suitable modifications of Θ' and Ψ'.

Let Θ''' be defined so that, for all σ,

$$\text{content}(\Theta'''(\sigma)) = \begin{cases} \emptyset, & \text{if content}(\Theta'(\sigma)) = \emptyset; \\ \{i+1\}, & \text{if content}(\Theta'(\sigma)) = \{i\} \text{ and } i > 1; \\ \{1\}, & \text{if } \{0\} \subseteq \text{content}(\Theta'(\sigma)) \subseteq \{0,1\} \\ & \text{and } S'_1 \not\subseteq \text{content}(\sigma); \\ \{2\}, & \text{if content}(\Theta'(\sigma)) = \{1\} \text{ and } S'_1 \subseteq \text{content}(\sigma); \\ \emptyset, & \text{if content}(\Theta'(\sigma)) = \{1\} \text{ and } S'_1 \not\subseteq \text{content}(\sigma); \\ \text{don't care}, & \text{otherwise.} \end{cases}$$

Where, the don't care entry means that we do not care what happens in this case. So let Θ''' be defined so that it is consistent with definition of Θ''' on initial segments of σ.

It is easy to verify that, for all $S \in \mathcal{S}$, if $\Theta'(S) = \{L_i\}$, then $\Theta'''(S) = \{L_{i+1}\}$.

We now describe Ψ'''. Let f be a recursive function such that

$$W_{f(n,g)} = \begin{cases} \{i-1\}, & \text{if } W_g^n = \{i\}, \text{ where } i > 1; \\ \{0,1\}, & \text{if } W_g^n = \{1\}; \\ \text{don't care}, & \text{otherwise.} \end{cases}$$

Let Ψ''' be defined so that $\Psi'''(g_0, g_1, \ldots) = \Psi'(f(0, g_0), f(1, g_1), \ldots)$. It is easy to verify that Θ''' and Ψ''' witness $\mathcal{S} \leq_{\text{strong}} \mathcal{L}$. ∎

Theorem 6 *For $i > 0$, let $L_i = \{i\}$. Let $L_0 = \{1, 0\}$. Let $\mathcal{L} = \{L_i \mid i > 0\}$. Let $\mathcal{L}' = \{L_0\} \cup \mathcal{L}$. (Note that $\mathcal{L} \leq_{\text{strong}} \mathcal{L}'$, but $\mathcal{L}' \not\leq_{\text{weak}} \mathcal{L}$). Then for all $\mathcal{S}$ such that $\mathcal{L} \leq_{\text{weak}} \mathcal{S} \leq_{\text{weak}} \mathcal{L}'$, either $\mathcal{S} \equiv_{\text{weak}} \mathcal{L}$ or $\mathcal{S} \equiv_{\text{weak}} \mathcal{L}'$.*

An easy modification of the proof for Theorem 5 can be used to show the above theorem. We omit the details.

FIN has been shown to be complete with respect to weak-reduction [JS94, JS96]. This means that *FIN* captures the essence of the most difficult learning problem with respect to weak-reduction. It was also shown that *FIN* is not complete with respect to strong-reduction [JS94, JS96]. Below we give an interesting collection of languages that is trivially identifiable (with 0 mind changes) but is not strong reducible to *FIN*.

Theorem 7 *Let* $\mathcal{L} = \{L \mid L \neq \emptyset \ \wedge \ (\forall x \in L)[W_x = L]\}$. *Then* $\mathcal{L} \not\leq_{\text{strong}} \mathit{FIN}$.

PROOF. Suppose by way of contradiction there exist enumeration operators Θ and Ψ that witness $\mathcal{L} \leq_{\text{strong}} \mathit{FIN}$. Then using the operator recursion theorem [Cas74], there exists a 1-1, increasing recursive function p such that languages, $W_{p(0)}, W_{p(1)}, W_{p(2)}, \ldots$, can be defined in stages ≥ 0, as follows.

Let $q_0 = 0$. Go to *Stage* 0.

begin *Stage* s

1. For $q_s \leq i \leq q_s + 2^s + 2$, let $W_{p(i)}$ enumerate $p(i)$.
2. For $q_s \leq i \leq q_s + 2^s + 2$, search for sequences τ_i such that
   content$(\tau_i) = \{p(i)\}$, and
   card$(\cup_{\{i \mid q_s \leq i \leq q_s + 2^s + 2\}}\text{content}(\Theta(\tau_i))) > s$.
3. If and when such τ_i's are found let $S = \{p(j) \mid j \leq q_s + 2^s + 2\}$.
4. For $j \leq q_s + 2^s + 2$, enumerate S in $W_{p(j)}$.
5. Let $q_{s+1} = q_s + 2^s + 3$.
6. Go to *Stage* $s + 1$.

end *Stage* s

There are two cases.

Case 1: Stage s starts but does not finish.

In this case, for $q_s \leq i \leq q_s + 2^s + 2$, $W_{p(i)} \in \mathcal{L}$. For $q_s \leq i \leq q_s + 2^s + 2$, let S_i be such that $\Theta(W_{p(i)}) = \{S_i\}$. Since the search in step 2 did not succeed, we have card$(\bigcup_{q_s \leq i \leq q_s + 2^s + 2} S_i) \leq s$. Now, since there are only 2^s distinct subsets of a set of size s, there exist distinct i and j such that $S_i = S_j$. But then $\Theta(W_{p(i)}) = \Theta(W_{p(j)})$, contradicting Lemma 1(c). Thus Θ, Ψ cannot witness $\mathcal{L} \leq_{\text{strong}} \mathit{FIN}$.

Case 2: Each stage halts.

Let $L = W_{p(0)}$. It is easy to see that for all i, $L = W_{p(i)} \in \mathcal{L}$. Suppose $\Theta(L) = \{S\}$. Now, by the success of step 2 in each stage s, we have that, card$(S) \geq s$, for all s. Hence S is infinite. Thus, Θ, Ψ do not witness $\mathcal{L} \leq_{\text{strong}} \mathit{FIN}$. ∎

5 Conclusion

The results presented in this paper describe the structure of the intrinsic complexity of language identification. It was shown that this structure contains an infinite hierarchy of language classes that represent learning problems of increasing difficulty. For $i > 1$, it was also shown that pattern languages are incomparable to FIN_i (language classes having $\leq i$ elements).

It was also shown that the structure of intrinsic complexity is rich as any finite directed acyclic graph can be embedded into the reducibility structure. It is open at this stage if any infinite directed acyclic graph can be embedded in the reducibility structure. It was also demonstrated that the reducibility structure is not dense.

It can be shown that the reducibility structure forms an upper semi lattice. To see this let E and O be mappings from 2^N into 2^N such that for $L \subseteq N$, $E(L) = \{2x \mid x \in L\}$ and $O(L) = \{2x+1 \mid x \in L\}$. For any two classes $\mathcal{L}_1$ and $\mathcal{L}_2$, it can be shown that $\{E(L) \mid L \in \mathcal{L}_1\} \cup \{O(L) \mid L \in \mathcal{L}_2\}$ is the least upper bound for $\mathcal{L}_1$ and $\mathcal{L}_2$ (with respect to both $\leq_{\text{weak}}$ and $\leq_{\text{strong}}$). It would be interesting to find out whether the reducibility structure forms a lattice.

It is felt that the results presented in this paper illustrate the intrinsic complexity of learning. Future work needs to concentrate on improving the fidelity of the operators so that a more illuminating structure can be brought to focus.

Acknowledgements

We thank the referee for helpful suggestions.

References

[Ang80a] D. Angluin. Finding patterns common to a set of strings. *Journal of Computer and System Sciences*, 21:46–62, 1980.

[Ang80b] D. Angluin. Inductive inference of formal languages from positive data. *Information and Control*, 45:117–135, 1980.

[Blu67] M. Blum. A machine independent theory of the complexity of recursive functions. *Journal of the ACM*, 14:322–336, 1967.

[Cas74] J. Case. Periodicity in generations of automata. *Mathematical Systems Theory*, 8:15–32, 1974.

[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.

[Fre91] R. Freivalds. Inductive inference of recursive functions: Qualitative theory. In J. Barzdins and D. Bjorner, editors, *Baltic Computer Science. Lecture Notes in Computer Science 502*, pages 77–110. Springer-Verlag, 1991.

[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.

[JS96] S. Jain and A. Sharma. On the intrinsic complexity of language identification. *Journal of Computer and System Sciences*, 1996. Accepted.

[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.