Language Learning from Texts: Degrees of Intrinsic Complexity and Their Characterizations

Sanjay Jain
School of Computing
National University of Singapore
Singapore 119260
sanjay@comp.nus.edu.sg

Efim Kinber
Department of Computer Science
Sacred Heart University
Fairfield, CT 06432-1000
kinbere@sacredheart.edu

Rolf Wiehagen
Department of Computer Science
University of Kaiserslautern
D-67653 Kaiserslautern
Germany
wiehagen@informatik.uni-kl.de

Abstract

This paper deals with two problems: 1) what makes languages to be learnable in the limit by natural strategies of varying hardness; 2) what makes classes of languages to be the hardest ones to learn. To quantify hardness of learning, we use *intrinsic complexity* based on reductions between learning problems. Two types of reductions are considered: weak reductions mapping texts (representations of languages) to texts, and strong reductions mapping languages to languages. For both types of reductions, characterizations of complete (hardest) classes in terms of their algorithmic and topological potentials have been obtained. To characterize the strong complete degree, we discovered a new and natural complete class capable of "coding" any learning problem using density of the set of rational numbers. We have also discovered and characterized rich hierarchies of degrees of complexity based on "core" natural learning problems. The classes in these hierarchies contain "multidimensional" languages, where the information learned from one dimension aids to learn other dimensions. In one formalization of this idea, the grammars learned from the dimensions $1, 2, \ldots, k$ specify the "subspace" for the dimension $k+1$, while the learning strategy for every dimension is predefined. In our other formalization, a "pattern" learned from the dimension k specifies the learning strategy for the dimension $k+1$. A number of open problems is discussed.

1 Introduction

There are two major objectives our paper attempts to achieve:

a) to discover what makes languages to be learnable in the limit by natural strategies of varying hardness;

b) to discover what makes classes of languages to be the hardest ones to learn.

The theory of learning languages in the limit, which has been quite advanced over the last three decades, suggests several ways to quantify hardness (complexity) of learning. The most popular among them are:

a) counting the number of *mind changes* [BF72, CS83, LZ93] the learner makes before arriving to the final hypothesis;

b) measuring the amount of (so-called *long-term*) memory the learner uses [Kin94, KS95];

c) reductions between different learning problems (classes of languages) and respective degrees of so-called *intrinsic* complexity [FKS95, JS96, JS97].

There have been several other notions of complexity of learning considered in the literature (for example see [Gol67, DS86, Wie86]).

The first two approaches above reveal quite interesting complexity hierarchies among learnable classes of languages ([CS83, LZ93, KS95]). However, a large number of interesting and very different natural classes of learnable classes falls into the category that requires more than uniformly bounded finite number of mind changes, as well as maximum (linear) amount of long-term memory. As it is demonstrated in our paper, intrinsic complexity of language learning, based on the idea of reductions, is perfectly suitable for quantifying hardness of many such natural classes of languages. It can be also successfully utilized to characterize the whole degrees of learnability based on these natural classes.

There are two different approaches to formalizing the concept of intrinsic complexity based on reductions between classes of languages [JS96]. In general terms, a major part of any reduction of one learning problem to another one is a mapping (an operator) that maps a language of the first learning problem to a language of the second one. A language is usually presented to a learner in form of a *text*, an infinite sequence of all elements of the language (possibly, with repetitions). Any non-empty language can be represented by many different texts. If a reduction may translate different texts of the *same* language to texts of *different* languages, we call such a reduction *weak*. If a reduction is required to translate all texts of the input language to texts of the *same* language, we call such a reduction *strong*. Roughly, a weak reduction translates texts to texts, while a strong reduction translates languages to languages. The paper [JS96] reveals significant differences between degrees of intrinsic complexity based on weak and, respectively, strong reductions.

For both types of reductions, we have obtained characterizations of complete degrees in terms of their algorithmic and topological potentials. For the case of strong reductions, we discovered a new natural complete class capable of "coding " (in the limit) any learning problem using density of the set of rational numbers. For weak reducibility, we were able to use the fact that the complete degree contains the class *FINITE* of all finite sets. The characterization for the weak complete degree is very different from any other characterization obtained in the paper - it is based on a requirement of density in terms of Baire topology. Note that a characterization of the complete degree of intrinsic complexity for *function* learning formulated in similar terms was obtained in [KPSW99]. The main difference between our characterization of weak complete degrees and the characterization for function learning in [KPSW99] is the requirement of *standardizability* (see Definition 5) for the hardest classes of languages. This notion, introduced quite long time ago in [Kin75, Fre91, JS94], for different purposes, turned out to be surprisingly useful for the characterization of all degrees in our paper.

For both types of reductions, we have also discovered and characterized rich structures of classes of languages, each of which requires its own specific type of learning strategy. Languages in these classes can be represented in "multidimensional" form, where the information obtained from learning one "dimension" aids in learning other "dimensions". We suggest and discuss several possibilities to formalize such "aid" and the ways it can be used. In the given paper, we concentrate on two following formalizations:

a) the grammars learned from the "dimensions" $L_1, L_2, \ldots, L_k$ specify the "subspace" containing the "sublanguage" L_{k+1};

b) the grammar learned from the "dimension" L_k codes a "pattern" that specifies a learning strategy for the class of languages containing L_{k+1}.

For the first formalization, we have obtained the complete picture of degrees of complexity for the classes of "multidimensional" languages based on combinations of probably the most important known natural classes of learnable languages: *INIT*, *COINIT*, *SINGLE*, *COSINGLE* (see Definition 6). Classes that can be defined under the second formalization turn out to be very complex. Yet we have shown that all of them are incomplete. The general problem whether such classes form a complexity hierarchy remains open.

In short, our major accomplishments are:

1) discovery of the fact that any language learning problem can be coded using sets $\{x \mid 0 \leq x \leq r\}$ of *rational* numbers;

2) characterizations of hardest learning problems in terms of their topological and algorithmic potentials;

3) discovery of a complex hierarchy of degrees of "multidimensional" languages; being interesting in its own right, this hierarchy can be used as a *scale* for quantifying hardness of learning complex concepts (for instance, it has been applied to quantify hardness of learning complex geometrical concepts in [JK99]).

Missing proofs, and some of the generalizations can be found in [JKW99].

2 Notation and Preliminaries

Any unexplained recursion theoretic notation is from [Rog67]. The symbol N denotes the set of natural numbers, $\{0, 1, 2, 3, \ldots\}$. Symbols $\emptyset$, $\subseteq$, $\subset$, $\supseteq$, and $\supset$ denote empty set, subset, proper subset, superset, and proper superset, respectively. $D_0, D_1, \ldots$, denotes a canonical recursive indexing of all the finite sets [Rog67, Page 70]. We assume that if $D_i \subseteq D_j$ then $i \leq j$ (the canonical indexing defined in [Rog67] satisfies this property). Cardinality of a set S is denoted by $\mathrm{card}(S)$. The maximum and minimum of a set are denoted by $\max(\cdot), \min(\cdot)$, respectively, where $\max(\emptyset) = 0$ and $\min(\emptyset) = \infty$. $L_1 \mathbf{\Delta} L_2$ denotes the symmetric difference of L_1 and L_2, that is $L_1 \mathbf{\Delta} L_2 = (L_1 - L_2) \cup (L_2 - L_1)$. For a natural number a, we say that $L_1 =^a L_2$, iff $\mathrm{card}(L_1 \mathbf{\Delta} L_2) \leq a$. We say that $L_1 =^* L_2$, iff $\mathrm{card}(L_1 \mathbf{\Delta} L_2) < \infty$. Thus, we take $n < * < \infty$, for all $n \in N$. If $L_1 =^a L_2$, then we say that L_1 is an a-variant of L_2.

We let $\langle \cdot, \cdot \rangle$ stand for an arbitrary, computable, bijective mapping from $N \times N$ onto N [Rog67]. We assume without loss of generality that $\langle \cdot, \cdot \rangle$ is monotonically increasing in both of its arguments. We define the corresponding projection functions: $\pi_1(\langle x, y \rangle) = x$ and $\pi_2(\langle x, y \rangle) = y$. $\langle \cdot, \cdot \rangle$ can be extended to n-tuples in a natural way (including $n = 1$, where $\langle x \rangle$ may be taken to be x). Projection functions $\pi_1, \ldots, \pi_n$ corresponding to n-tuples can be defined similarly (where the tuple size would be clear from context). Due to the above isomorphism between N^k and N, we often identify the tuple $(x_1, \cdots, x_n)$ with $\langle x_1, \cdots, x_n \rangle$.

By φ we denote a fixed *acceptable* programming system for the partial computable functions mapping N to N [Rog67, MY78]. By φ_i we denote the partial computable function computed by the program with number i in the φ-system. Symbol $\mathcal{R}$ denotes the set of all recursive functions, that is total computable functions. By Φ we denote an arbitrary fixed Blum complexity measure [Blu67, HU79] for the φ-system. By W_i we denote $\mathrm{domain}(\varphi_i)$. 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. Symbol L, with or without decorations, ranges over $\mathcal{E}$. By $\overline{L}$, we denote the complement of L, that is $N - L$. Symbol $\mathcal{L}$, with or without decorations, ranges over subsets of $\mathcal{E}$. By $W_{i,s}$ we denote the set $\{x < s \mid \Phi_i(x) < s\}$.

A class $\mathcal{L} \subseteq \mathcal{E}$ is said to be recursively enumerable (r.e.) [Rog67], iff $\mathcal{L} = \emptyset$ or there exists a recursive function f such that $\mathcal{L} = \{W_{f(i)} \mid i \in N\}$. In this latter case we say that $W_{f(0)}, W_{f(1)}, \ldots$ is a recursive enumeration of $\mathcal{L}$. $\mathcal{L}$ is said to be 1–1 enumerable iff (i) $\mathcal{L}$ is finite or (ii) there exists a recursive function f such that $\mathcal{L} = \{W_{f(i)} \mid i \in N\}$ and $W_{f(i)} \neq W_{f(j)}$, if $i \neq j$. In this latter case we say that $W_{f(0)}, W_{f(1)}, \ldots$ is a 1–1 recursive enumeration of $\mathcal{L}$.

A partial function F from N to N is said to be partial limit recursive, iff there exists a recursive function f from $N \times N$ to N such that for all x, $F(x) = \lim_{y\to\infty} f(x,y)$. Here if $F(x)$ is not defined then $\lim_{y\to\infty} f(x,y)$, must also be undefined. A partial limit recursive function F is called (total) limit recursive function, if F is total. $\downarrow$ denotes defined or converges. $\uparrow$ denotes undefined or diverges.

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

Definition 1 (a) A *sequence* σ is a mapping from an initial segment of N into $(N \cup \{\#\})$. The empty sequence is denoted by Λ.

(b) The *content* of a sequence σ, denoted content(σ), 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, with or without decorations, range over finite sequences. SEQ denotes the set of all finite sequences.

Definition 2 [Gol67] (a) 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.

(b) The *content* of a text T, denoted by content(T), is the set of natural numbers in the range of T; that is, the language which T is a text for.

(c) $T[n]$ denotes the finite initial sequence of T with length n.

We let T, with or without decorations, range over texts. We let $\mathcal{T}$ range over sets of texts.

A class $\mathcal{T}$ of texts is said to be r.e. iff there exists a recursive function f, and a sequence $T_0, T_1, \ldots$ of texts such that $\mathcal{T} = \{T_i \mid i \in N\}$, and, for all i, x, $T_i(x) = f(i,x)$.

Definition 3 A *language learning machine* [Gol67] is an algorithmic device which computes a mapping from SEQ into N.

We let $\mathbf{M}$, with or without decorations, range over learning machines. $\mathbf{M}(T[n])$ is interpreted as the grammar (index for an accepting program) conjectured by the learning machine $\mathbf{M}$ on the initial sequence $T[n]$. We say that $\mathbf{M}$ converges on T to i, (written $\mathbf{M}(T)\downarrow = i$) iff $(\stackrel{\infty}{\forall} n)[\mathbf{M}(T[n]) = i]$.

There are several criteria for a learning machine to be successful on a language. Below we define identification in the limit introduced by Gold [Gol67].

Definition 4 [Gol67, CS83] Suppose $a \in N \cup \{*\}$.

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

(b) $\mathbf{M}$ $\mathbf{TxtEx}^a$-identifies an r.e. language L (written: $L \in \mathbf{TxtEx}^a(\mathbf{M})$) just in case $\mathbf{M}$ $\mathbf{TxtEx}^a$-identifies each text for L.

(c) $\mathbf{M}$ $\mathbf{TxtEx}^a$-identifies a class $\mathcal{L}$ of r.e. languages (written: $\mathcal{L} \subseteq \mathbf{TxtEx}^a(\mathbf{M})$) just in case $\mathbf{M}$ $\mathbf{TxtEx}^a$-identifies each language from $\mathcal{L}$.

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

For $a = 0$, we often write $\mathbf{TxtEx}$ instead of $\mathbf{TxtEx}^0$.

Other criteria of success are finite identification [Gol67], behaviorally correct identification [Fel72, OW82, CL82], and vacillatory identification [OW82, Cas88]. In the present paper, we only discuss results about $\mathbf{TxtEx}^a$-identification.

The following definition is a generalization of the definition of limiting standardizability considered in [Kin75, Fre91, JS94].

Definition 5 Let $a \in N \cup \{*\}$. A class $\mathcal{L}$ of recursively enumerable sets is called *a-limiting standardizable* iff there exists a partial limiting recursive function F such that

(a) For all i such that $W_i =^a L$ for some $L \in \mathcal{L}$, $F(i)$ is defined.

(b) For all $L, L' \in \mathcal{L}$, for all i, j such that $W_i =^a L$ and $W_j =^a L'$,

$$F(i) = F(j) \Leftrightarrow L = L'.$$

[Kin75, Fre91, JS94] $\mathcal{L}$ is called *limiting standardizable* iff $\mathcal{L}$ is 0-limiting standardizable.

Thus, informally, a class $\mathcal{L}$ of r.e. languages is limiting standardizable if all the infinitely many grammers $i \in N$ of each language $L \in \mathcal{L}$ can be mapped ("standardized") in the limit to some unique grammar (natural number). Notice that it is not required that this "standard grammar" must be a grammar of L again. However, standard grammars for *different* languages from $\mathcal{L}$ have to be pairwise different.

The following basic classes of languages will be used frequently in the following.

Definition 6 $\mathit{SINGLE} = \{L \mid (\exists i)[L = \{i\}]\}$.
$\mathit{COSINGLE} = \{L \mid (\exists i)[L = N - \{i\}]\}$.

$INIT = \{L \mid (\exists i)[L = \{x \mid x \leq i\}]\}$.
$COINIT = \{L \mid (\exists i)[L = \{x \mid x \geq i\}]\}$.
$FINITE = \{L \mid L \textit{ is a finite subset of } N\}$.

3 Weak and Strong Reductions

We first present some technical machinery.

We write $\sigma \subseteq \tau$ if σ is an initial segment of τ, and $\sigma \subset \tau$ if σ is a proper initial segment 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 $n \in N$, $\sigma^n = T[|\sigma^n|]$. This text is denoted by $\bigcup_n \sigma^n$. Let $\mathbf{T}$ denote the set of all texts, that is, the set of all infinite sequences over $N \cup \{\#\}$.

We define an *enumeration operator* (or just operator), Θ, to be an algorithmic mapping from SEQ into SEQ such that for all $\sigma, \tau \in$ SEQ, if $\sigma \subseteq \tau$, then $\Theta(\sigma) \subseteq \Theta(\tau)$. We further assume that for all texts T, $\lim_{n\to\infty} |\Theta(T[n])| = \infty$. By extension, we think of Θ as also defining a mapping from $\mathbf{T}$ into $\mathbf{T}$ such that $\Theta(T) = \bigcup_n \Theta(T[n])$.

A final notation about the operator Θ. If for a language L, there exists an L' such that for each text T for L, $\Theta(T)$ is a text for L', then we write $\Theta(L) = L'$, else we say that $\Theta(L)$ is undefined. 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 $\Theta(\mathcal{T}) = \{\Theta(T) \mid T \in \mathcal{T}\}$, and $\Theta(\mathcal{L}) = \{\Theta(L) \mid L \in \mathcal{L}\}$.

We also need the notion of an infinite sequence of grammars. We let α, with or without decorations, range over infinite sequences of grammars. From the discussion in the previous section it is clear that 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 $\alpha = i_0, i_1, i_2, i_3, \ldots$, then $\alpha[3]$ denotes the sequence i_0, i_1, i_2, and $\alpha(3)$ is i_3. Furthermore, we say that α converges to i if there exists an n such that, for all $n' \geq n$, $i_{n'} = i$.

Let $\mathbf{I}$ be any criterion for language identification from texts, for example $\mathbf{I} = \mathbf{TxtEx}^a$. We say that an infinite sequence α of grammars is $\mathbf{I}$-*admissible* for text T just in case α witnesses $\mathbf{I}$-identification of text T. So, if $\alpha = i_0, i_1, i_2, \ldots$ is a $\mathbf{TxtEx}^a$-admissible sequence for T, then α converges to some i such that $W_i =^a$ content(T); that is, the limit i of the sequence α is a grammar for an a-variant of the language content(T).

We now formally introduce our reductions. Although in this paper we will only be concerned with $\mathbf{TxtEx}^a$-identification, we present the general case of the definition.

Definition 7 [JS96] Let $\mathcal{L}_1 \subseteq \mathcal{E}$ and $\mathcal{L}_2 \subseteq \mathcal{E}$ be given. Let identification criteria $\mathbf{I}_1$ and $\mathbf{I}_2$ be given. Let $\mathcal{T}_1 = \{T \mid T \text{ is a text for } L \in \mathcal{L}_1\}$. Let $\mathcal{T}_2 = \{T \mid T \text{ is a text for } L \in \mathcal{L}_2\}$. We say that $\mathcal{L}_1 \leq^{\mathbf{I}_1,\mathbf{I}_2}_{\text{weak}} \mathcal{L}_2$ just in case there exist operators Θ and Ψ such that for all $T \in \mathcal{T}_1$ and for all infinite sequences α of grammars the following hold:

(a) $\Theta(T) \in \mathcal{T}_2$ and

(b) if α is an $\mathbf{I}_2$-admissible sequence for $\Theta(T)$, then $\Psi(\alpha)$ is an $\mathbf{I}_1$-admissible sequence for T.

We say that $\mathcal{L}_1 \leq^{\mathbf{I}}_{\text{weak}} \mathcal{L}_2$ iff $\mathcal{L}_1 \leq^{\mathbf{I},\mathbf{I}}_{\text{weak}} \mathcal{L}_2$. We say that $\mathcal{L}_1 \equiv^{\mathbf{I}}_{\text{weak}} \mathcal{L}_2$ iff $\mathcal{L}_1 \leq^{\mathbf{I}}_{\text{weak}} \mathcal{L}_2$ and $\mathcal{L}_2 \leq^{\mathbf{I}}_{\text{weak}} \mathcal{L}_1$.

Intuitively, $\mathcal{L}_1 \leq^{\mathbf{I}}_{\text{weak}} \mathcal{L}_2$ just in case there exists an operator Θ that transforms texts for languages in $\mathcal{L}_1$ into texts for languages in $\mathcal{L}_2$ and there exists another operator Ψ that behaves as follows: if Θ transforms text T (for a language in $\mathcal{L}_1$) to text T' (for a language in $\mathcal{L}_2$), then Ψ transforms $\mathbf{I}$-admissible sequences for T' into $\mathbf{I}$-admissible sequences for T.

For many commonly studied criteria of inference, such as $\mathbf{I} = \mathbf{TxtEx}^a$, if $\mathcal{L}_1 \leq^{\mathbf{I}}_{\text{weak}} \mathcal{L}_2$ then, intuitively, the problem of identifying $\mathcal{L}_2$ in the sense of $\mathbf{I}$ is at least as hard as the problem of identifying $\mathcal{L}_1$ in the sense of $\mathbf{I}$, since the solvability of the former problem implies the solvability of the latter one. That is, given any machine $\mathbf{M}_2$ which $\mathbf{I}$-identifies $\mathcal{L}_2$, it is easy to construct a machine $\mathbf{M}_1$ which $\mathbf{I}$-identifies $\mathcal{L}_1$. To see this for $\mathbf{I} = \mathbf{TxtEx}^a$, suppose Θ and Ψ witness $\mathcal{L}_1 \leq^{\mathbf{I}}_{\text{weak}} \mathcal{L}_2$. $\mathbf{M}_1(T)$, for a text T is defined as follows. Let $p_n = \mathbf{M}_2(\Theta(T)[n])$, and $\alpha = p_0, p_1, \ldots$. Let $\alpha' = \Psi(\alpha) = p'_0, p'_1, \ldots$. Then let $\mathbf{M}_1(T) = \lim_{n\to\infty} p'_n$. Consequently, $\mathcal{L}_2$ may be considered as a "hardest" problem for $\mathbf{I}$-identification if for *all* classes $\mathcal{L}_1 \in \mathbf{I}$, $\mathcal{L}_1 \leq^{\mathbf{I}}_{\text{weak}} \mathcal{L}_2$ holds. If $\mathcal{L}_2$ itself belongs to $\mathbf{I}$, then $\mathcal{L}_2$ is said to be *complete*. We now formally define these notions of hardness and completeness for the above reduction.

Definition 8 [JS96] Let $\mathbf{I}$ be an identification criterion. Let $\mathcal{L} \subseteq \mathcal{E}$ be given.

(a) If for all $\mathcal{L}' \in \mathbf{I}$, $\mathcal{L}' \leq^{\mathbf{I}}_{\text{weak}} \mathcal{L}$, then $\mathcal{L}$ is $\leq^{\mathbf{I}}_{\text{weak}}$-*hard*.

(b) If $\mathcal{L}$ is $\leq^{\mathbf{I}}_{\text{weak}}$-hard and $\mathcal{L} \in \mathbf{I}$, then $\mathcal{L}$ is $\leq^{\mathbf{I}}_{\text{weak}}$-*complete*.

It should be noted that if $\mathcal{L}_1 \leq^{\mathbf{I}}_{\text{weak}} \mathcal{L}_2$ by operators Θ and Ψ, then there is no requirement that Θ maps all texts for each 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 [JS96] Let $\mathcal{L}_1 \subseteq \mathcal{E}$ and $\mathcal{L}_2 \subseteq \mathcal{E}$ be given. We say that $\mathcal{L}_1 \leq^{\mathbf{I}_1,\mathbf{I}_2}_{\text{strong}} \mathcal{L}_2$ just in case there exist operators Θ, Ψ witnessing that $\mathcal{L}_1 \leq^{\mathbf{I}_1,\mathbf{I}_2}_{\text{weak}} \mathcal{L}_2$, and for all $L_1 \in \mathcal{L}_1$, there exists an $L_2 \in \mathcal{L}_2$, such that $(\forall \text{ texts } T \text{ for } L_1)[\Theta(T) \text{ is a text for } L_2]$.

We say that $\mathcal{L}_1 \leq^{\mathbf{I}}_{\text{strong}} \mathcal{L}_2$ iff $\mathcal{L}_1 \leq^{\mathbf{I},\mathbf{I}}_{\text{strong}} \mathcal{L}_2$. We say that $\mathcal{L}_1 \equiv^{\mathbf{I}}_{\text{strong}} \mathcal{L}_2$ iff $\mathcal{L}_1 \leq^{\mathbf{I}}_{\text{strong}} \mathcal{L}_2$ and $\mathcal{L}_2 \leq^{\mathbf{I}}_{\text{strong}} \mathcal{L}_1$.

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

It is easy to see that $\leq^{\mathbf{TxtEx}^a}_{\text{weak}}$ and $\leq^{\mathbf{TxtEx}^a}_{\text{strong}}$ are reflexive and transitive, and that $\mathcal{L} \leq^{\mathbf{TxtEx}^a}_{\text{strong}} \mathcal{L}'$ implies $\mathcal{L} \leq^{\mathbf{TxtEx}^a}_{\text{weak}} \mathcal{L}'$.

Proposition 1 *(based on [JS97]) Suppose $\mathcal{L} \leq^{\mathbf{I}}_{\text{strong}} \mathcal{L}'$, via Θ and Ψ. Then, for all $L, L' \in \mathcal{L}$, $L \subseteq L' \Rightarrow \Theta(L) \subseteq \Theta(L')$.*

We will be using Proposition 1 implicitly when we are dealing with strong reductions. Since, for $\mathcal{L} \leq^{\mathbf{I}}_{\text{strong}} \mathcal{L}'$ via Θ and Ψ, for all $L \in \mathcal{L}$, $\Theta(L)$ is defined (= some $L' \in \mathcal{L}'$), when considering strong reductions, we often consider Θ as mapping sets to sets instead of mapping sequences to sequences. This is clearly without loss of generality, as one can easily convert such Θ to Θ as in Definition 9 of strong-reduction.

4 A Natural Strongly Complete Class and a Characterization of Strongly Complete Classes

In this section we exhibit a natural class which is $\leq^{\mathbf{TxtEx}^a}_{\text{strong}}$-complete for all $a \in N$ (see Theorem 2). Corollary 1 to Theorem 2 then shows an even simpler class, $RINIT_{0,1}$ defined below, as $\leq^{\mathbf{TxtEx}}_{\text{strong}}$-complete. We also characterize the $\leq^{\mathbf{TxtEx}^a}_{\text{strong}}$-complete degree, for all $a \in N$, in Theorem 3.

Let **rat** denote the set of all non-negative rational numbers. For $s, r \in \mathbf{rat}$, let $\mathbf{rat}_{s,r} = \{x \in \mathbf{rat} \mid s \leq x \leq r\}$. For allowing us to consider r.e. sets of rational numbers, let coderat$(\cdot)$ denote an effective bijective mapping from **rat** to N.

Definition 10 Suppose $r \in \mathbf{rat}_{0,1}$.

Let $X_r = \{\text{coderat}(x) \mid x \in \mathbf{rat} \text{ and } 0 \leq x \leq r\}$.

Let $X^{cyl}_r = \{\text{coderat}(2w + x) \mid x \in \mathbf{rat},\ w \in N \text{ and } 0 \leq x \leq r\}$.

Definition 11 Suppose $s, r \in \mathbf{rat}_{0,1}$ and $s < r$.

Let $RINIT_{s,r} = \{X_w \mid w \in \mathbf{rat}_{s,r}\}$.

Let $RINIT^{cyl}_{s,r} = \{X^{cyl}_w \mid w \in \mathbf{rat}_{s,r}\}$.

Our main goal in this section is to show that the class $RINIT_{0,1}$ is complete. Informally, we have to demonstrate that every language learning problem can be effectively coded as a sequence of increasing rationals that stabilizes to one rational in the interval $[0,1]$. More specifically, we code by rationals the sequence of hypotheses outputted by a (modified) learning device being fed an arbitrary text of a learnable language. First, we prove a simple technical Proposition 2 that gives us opportunity to algorithmically generate sequences of rationals that tend to get closer to each other still keeping previously chosen distances between them; these sequences are necessary for coding. Using Theorem 1 gives us opportunity to use learning machines **M** that have special properties: their outputs do not depend on arrangement and order of language elements in the input. Using such a machine Proposition 5 allows us to construct a "learning device" H that stabilizes its conjectures on certain "full locking sequences" for the underlying languages. Using the functions provided by Proposition 2, one can map sequences of conjectures produced by H on inputs stabilizing to "full locking sequences" to sequences of rationals stabilizing to a rational representing a language in $RINIT_{0,1}$.

In some cases below, in the pairing function we will be using finite sets as arguments (for example $\langle S, l\rangle$). This is for ease of notation: $\langle S, l\rangle$ should be understood as $\langle x, l\rangle$, where x is a canonical code [Rog67] for the finite set S (i.e. $D_x = S$).

Proposition 2 *There exist recursive functions F and ϵ from $\mathbf{rat}_{0,1}$ to $\mathbf{rat}_{0,1}$ such that, for all rationals, x, y, where $0 \leq x < y \leq 1$,*

$$F(x) + \epsilon(x) < F(y).$$

Moreover, $F(1) + \epsilon(1) \leq 1$.

PROOF. Let $q_0, q_1, \ldots,$ be some 1–1 recursive enumeration of all the rational numbers between 0 and 1 (both inclusive), such that $q_0 = 0$ and $q_1 = 1$.

We define, inductively on i, $F(q_i)$ and $\epsilon(q_i)$.

Let $F(0) = 1/8$ and $\epsilon(0) = 1/8$. Let $F(1) = 7/8$, $\epsilon(1) = 1/8$.

Induction Hypothesis: Suppose we have defined $F(q_i)$ and $\epsilon(q_i)$, for $i \leq k$. Then for all $j, j' \leq k$, $[q_j < q_{j'} \Rightarrow F(q_j) + \epsilon(q_j) < F(q_{j'})]$. Note that the induction hypothesis is clearly true for $k = 1$.

Now suppose that $F(q_i)$ and $\epsilon(i)$ have been defined for $i \leq k$.

We now define $F(q_{k+1})$ and $\epsilon(q_{k+1})$ as follows.

Let $p_1 = \max(\{q_i \mid i \leq k \ \wedge\ q_i < q_{k+1}\})$. Let $p_2 = \min(\{q_i \mid i \leq k \ \wedge\ q_i > q_{k+1}\})$.

By induction hypothesis, $F(p_1) + \epsilon(p_1) < F(p_2)$.

Let $F(q_{k+1}) = F(p_1) + \epsilon(p_1) + [F(p_2) - (F(p_1) + \epsilon(p_1))]/3$, and $\epsilon(q_{k+1}) = [F(p_2) - (F(p_1) + \epsilon(p_1))]/3$.

It is easy to verify that the induction hypothesis is satisfied. The proposition follows. ∎

Fix F, ϵ as in the above proposition.

For $S \in FINITE$, let $code(S) = \sum_{x \in S} 2^{-x-1}$. Note that $0 \leq code(S) < 1$.

Note that, if $\min(S - S') < \min(S' - S)$, then $code(S) > code(S')$ (here $\min(\emptyset) = \infty$).

For $S \in FINITE$ and $l \in N$, let $G(\langle S, l\rangle) = F(code(S)) + \epsilon(code(S)) - \frac{\epsilon(code(S))}{l+2}$.

Proposition 3 *G is a recursive mapping from N to $\mathbf{rat}_{0,1}$. Moreover, if $\min(S - S') < \min(S' - S)$ or $S = S'$ and $l > l'$, then $G(\langle S, l\rangle) > G(\langle S', l'\rangle)$.*

PROOF. Follows from definition of G. ∎

Definition 12 [Ful90, BB75] A machine $\mathbf{M}$ is said to be *rearrangement independent* iff for all $\sigma, \tau \in$ SEQ, if content(σ) = content(τ), and $|\sigma| = |\tau|$, then $\mathbf{M}(\sigma) = \mathbf{M}(\tau)$.

A machine $\mathbf{M}$ is said to be *order independent* iff for all texts T and T', if content(T) = content(T'), then either both $\mathbf{M}(T)$ and $\mathbf{M}(T')$ are undefined, or both are defined and $\mathbf{M}(T) = \mathbf{M}(T')$.

Note that rearrangement independent machines base their output only on the content and length of the input. Thus for $l \geq \text{card}(S)$, we define $\beta^{S,l}$ as the lexicographically least σ of length l such that content$(\sigma) = S$.

Theorem 1 *(based on [Ful90]) Suppose $a \in N \cup \{*\}$ and $\mathcal{L} \in \mathbf{TxtEx}^a$. Then there exists a rearrangement independent and order independent machine $\mathbf{M}$ such that $\mathcal{L} \subseteq \mathbf{TxtEx}^a(\mathbf{M})$.*

Definition 13 [Ful90, BB75] $\sigma \in$ SEQ is said to be a *stabilizing sequence* for $\mathbf{M}$ on L, iff content$(\sigma) \subseteq L$, and for all τ such that $\sigma \subseteq \tau$ and content$(\tau) \subseteq L$, $\mathbf{M}(\sigma) = \mathbf{M}(\tau)$.

$\sigma \in$ SEQ is said to be a $\mathbf{TxtEx}^a$*-locking sequence* for $\mathbf{M}$ on L, iff σ is a stabilizing sequence for $\mathbf{M}$ on L, and $W_{\mathbf{M}(\sigma)} =^a L$.

Lemma 1 *(based on [BB75, JORS99]) Suppose $a \in N \cup \{*\}$. If $\mathbf{M}$ $\mathbf{TxtEx}^a$-identifies L, then there exists a stabilizing sequence for $\mathbf{M}$ on L, and every stabilizing sequence for M on L is a $\mathbf{TxtEx}^a$-locking sequence for $\mathbf{M}$ on L.*

Definition 14 Suppose $\mathbf{M}$ is a rearrangement independent and order independent learning machine. Let $S \in$ *FINITE* and $l \in N$.
(a) $\langle S, l\rangle$ is said to be a *full-stabilizing-sequence* for $\mathbf{M}$ on L iff:
(i) $l > \max(S)$,
(ii) $(\forall x < l)[x \in L \Leftrightarrow x \in S]$,
(iii) $\beta^{S,2l}$ is a stabilizing sequence for $\mathbf{M}$ on L.
(b) Suppose $a \in N \cup \{*\}$. $\langle S, l\rangle$ is said to be a $\mathbf{TxtEx}^a$-*full-locking-sequence* for $\mathbf{M}$ on L, iff $\langle S, l\rangle$ is a full-stabilizing-sequence for $\mathbf{M}$ on L, and $W_{\mathbf{M}(\beta^{S,2l})} =^a L$.

Intuitively, $\langle S, l\rangle$ is a full-stabilizing-sequence ($\mathbf{TxtEx}^a$-full-locking-sequence) for $\mathbf{M}$ on L, if $\beta^{S,2l}$ is a stabilizing sequence ($\mathbf{TxtEx}^a$-locking sequence) for $\mathbf{M}$ on L, and $\beta^{S,2l}$ contains exactly the elements in L which are less than l.

Proposition 4 *Suppose $a \in N \cup \{*\}$ and $\mathbf{M}$ is a rearrangement independent and order independent machine, which $\mathbf{TxtEx}^a$-identifies L. Then there exists a full-stabilizing-sequence for $\mathbf{M}$ on L. Moreover, every full-stabilizing-sequence for $\mathbf{M}$ on L is a $\mathbf{TxtEx}^a$-full-locking-sequence for $\mathbf{M}$ on L.*

PROOF. Suppose $\mathbf{M}$ $\mathbf{TxtEx}^a$-identifies L. Suppose σ is a stabilizing-sequence for $\mathbf{M}$ on L. Let $l = 1 + \max(\{|\sigma|\} \cup \text{content}(\sigma))$, and $S = \{x \mid x < l \wedge x \in L\}$. It follows that $\beta^{S,2l}$ is also a stabilizing-sequence for $\mathbf{M}$ on L. Thus, $\langle S, l\rangle$ is a full-stabilizing-sequence for $\mathbf{M}$ on L. The second part of the proposition follows from Lemma 1. ∎

Definition 15 We say that $\langle S, l\rangle$ is the *least full-stabilizing-sequence* for $\mathbf{M}$ on L, iff $\langle S, l\rangle$ is a full-stabilizing-sequence for $\mathbf{M}$ on L which minimizes l.

Proposition 5 *Suppose $\mathbf{M}$ is a rearrangement independent and order independent machine. Then, there exists a recursive function H mapping SEQ to N, such that*

(i) For all $\sigma \in$ SEQ, if $H(\sigma) = \langle S, l\rangle$, then $\max(S) < l$.

(ii) For all $\sigma \subseteq \tau$, $G(H(\tau)) \geq G(H(\sigma))$.

(iii) For all texts T, $H(T) = \lim_{n\to\infty} H(T[n])$ converges to the least full-stabilizing-sequence for $\mathbf{M}$ on content(T)*, if any.*

PROOF. Define $H(\sigma)$ as follows:

For $l \leq 1 + \max(\text{content}(\sigma) \cup \{|\sigma|\})$, let $S^\sigma_l = \text{content}(\sigma) \cap \{x \mid x < l\}$.

Let $H(\sigma) = \langle S^\sigma_l, l\rangle$, for the least $l \leq 1 + \max(\text{content}(\sigma) \cup \{|\sigma|\})$, such that

$$(\forall \tau \mid \beta^{S^\sigma_l, 2l} \subseteq \tau \wedge \text{content}(\tau) \subseteq \text{content}(\sigma) \wedge |\tau| \leq |\sigma|)$$

$$[\mathbf{M}(\beta^{S^\sigma_l, 2l}) = \mathbf{M}(\tau)]$$

Note that there exists an l as above, since $l = 1 + \max(\text{content}(\sigma) \cup \{|\sigma|\})$, satisfies the requirements.

Using Proposition 3, we claim that H satisfies the properties above. (i) is trivially true. Clearly, $H(T)$ converges to the least full-stabilizing-sequence for $\mathbf{M}$ on content(T), if any. Thus, (iii) is satisfied. Now we consider the monotonicity requirement (ii). Suppose $\sigma \subseteq \tau$. Suppose $H(\sigma) = \langle S^\sigma_l, l\rangle$ and $H(\tau) = \langle S^\tau_{l'}, l'\rangle$.

(1) Clearly, $S^\sigma_w \subseteq S^\tau_w$, for all w.

(2) If $l' < l$, then $S^\tau_{l'}$ must be a proper superset of $S^\sigma_{l'}$ (otherwise $\langle S^\tau_{l'}, l'\rangle$ would have been a candidate for consideration as full-stabilizing-sequence even for input σ). Thus, $G(\langle S^\tau_{l'}, l'\rangle) > G(\langle S^\sigma_l, l\rangle)$, by Proposition 3.

(3) If $l' \geq l$, then $S^\tau_{l'} \supseteq S^\sigma_l$. Thus, $G(\langle S^\tau_{l'}, l'\rangle) \geq G(\langle S^\sigma_l, l\rangle)$, by Proposition 3. ∎

Theorem 2 *For any $a \in N$, $RINIT^{cyl}_{0,1}$ is $\leq^{\mathbf{TxtEx}^a}_{\text{strong}}$-complete.*

Proof. Clearly $RINIT^{cyl}_{0,1} \in \mathbf{TxtEx} \subseteq \mathbf{TxtEx}^a$.

Suppose $\mathcal{L} \in \mathbf{TxtEx}^a$. Let $\mathbf{M}$ be a rearrangement independent and order independent machine which $\mathbf{TxtEx}^a$-identifies $\mathcal{L}$.

Let H be as in Proposition 5.

Let Θ be defined as follows.

Let $\Theta(\sigma) = X^{cyl}_{G(H(\sigma))}$. Note that for $L \in \mathbf{TxtEx}^a(\mathbf{M})$, $\Theta(L) = X^{cyl}_{G(\langle S,l\rangle)}$, where $\langle S,l\rangle$, is the least full-stabilizing-sequence for $\mathbf{M}$ on L (by Proposition 5).

Ψ is defined as follows. Suppose a sequence α of grammars converges to a grammar p. (If there is no such p, then it does not matter what Ψ outputs on sequence α). Suppose $x \in \mathbf{rat}_{0,1}$ is the maximum rational number (if any) such that $\text{coderat}(2w + x) \in W_p$, for at least $2a+1$ different $w \in N$. (If there is no such x, then it does not matter what Ψ outputs on sequence α). Suppose $S \in FINITE, l \in N$ (if any) are such that $x = G(\langle S,l\rangle)$. (If there are no such S, l, then it does not matter what Ψ outputs on sequence α). Then, $\Psi(\alpha)$ converges to $\mathbf{M}(\beta^{S,2l})$. It is easy to verify that Θ and Ψ witness that $\mathbf{TxtEx}^a(\mathbf{M}) \leq^{\mathbf{TxtEx}^a}_{\text{strong}} RINIT^{cyl}_{0,1}$.

This completes the proof of Theorem 2. ∎

Corollary 1 *$RINIT_{0,1}$ is $\leq^{\mathbf{TxtEx}}_{\text{strong}}$-complete.*

Why $RINIT_{0,1}$ is complete and, say, $INIT$ is not? From the first glance, strategies learning both classes seem to be identical: being fed the input text, pick the largest number in it to represent the language to be learned. However, there is a subtle difference. Numbers in any language in $INIT$ can be listed in the ascending order, while for the rationals in languages from $RINIT_{0,1}$ it is not possible. Learning, say, the language $\{0, 1, 2, 3, 4, 5, 6\}$, being fed the number 3, we need at most three "mind changes" to arrive at the correct hypothesis. On the other hand, learning the language $X_{2/3}$, we always choose the largest number in the input as our conjecture, however, 1/2 being such a number in the initial fragment of the input does not impact in any way the number of mind changes that will yet occur before we arrive at the final conjecture 2/3 – it depends entirely on the input. This lack of any conceivable bound on the number of remaining mind changes differentiates $RINIT_{0,1}$ from all other, non-complete, classes observed in our paper.

Theorem 3 *For any $a \in N$ and any $\mathcal{L} \in \mathbf{TxtEx}^a$, $\mathcal{L}$ is $\leq^{\mathbf{TxtEx}^a}_{\text{strong}}$-complete iff there exists a recursive function H from $\mathbf{rat}_{0,1}$ to N such that:*

(a) $\{W_{H(r)} \mid r \in \mathbf{rat}_{0,1}\} \subseteq \mathcal{L}$.

(b) If $0 \leq r < r' \leq 1$, then $W_{H(r)} \subset W_{H(r')}$.

(c) $\{W_{H(r)} \mid r \in \mathbf{rat}_{0,1}\}$ is a-limiting standardizable.

Proof. For the whole proof, for $q \in \mathbf{rat}_{0,1}$, let T_q denote a text, obtained effectively from q, for X^{cyl}_q.

Necessity. Using Theorem 2, suppose $RINIT^{cyl}_{0,1} \leq^{\mathbf{TxtEx}^a}_{\text{strong}} \mathcal{L}$ via Θ, Ψ.

Define H and E as follows.

$W_{H(q)} = \text{content}(\Theta(T_q))$, for $q \in \mathbf{rat}_{0,1}$.

E defined below will witness the a-limiting standardizability of $\{W_{H(r)} \mid r \in \mathbf{rat}_{0,1}\}$. $E(p)$ is defined as follows. Suppose $\alpha_p = p, p, p, \ldots$. Suppose $\Psi(\alpha_p)$ converges to w. Then $E(p) =$ maximum rational number $r \in \mathbf{rat}_{0,1}$ (if any) such that, for at least $2a+1$ different natural numbers m, $\text{coderat}(2m + r) \in W_w$.

It is easy to verify that H satisfies parts (a) and (b) of the theorem and E witnesses the a-limiting standardizability as required in part (c).

Sufficiency. Suppose that H is as given in the theorem, and E witnesses the a-limiting standardizability as given in condition (c) of the theorem.

Then, define Θ and Ψ witnessing $RINIT^{cyl}_{0,1} \leq^{\mathbf{TxtEx}^a}_{\text{strong}} \mathcal{L}$ as follows:

$\Theta(L) = \bigcup\{W_{H(q)} \mid \text{coderat}(q) \in L \ \wedge \ q \in \mathbf{rat}_{0,1}\}$.

Let p_q denote a grammar (obtained effectively from q), for $\text{content}(\Theta(T_q))$.

Define Ψ as follows. Suppose a sequence α of grammars converges to a grammar i. Then, $\Psi(\alpha)$ converges to a grammar for X^{cyl}_q, such that $E(i) = E(p_q)$ (if there is any such $q \in \mathbf{rat}_{0,1}$).

It is easy to verify that Θ and Ψ witness that $RINIT^{cyl}_{0,1} \leq^{\mathbf{TxtEx}^a}_{\text{strong}} \mathcal{L}$.

Hence $\mathcal{L}$ is $\leq^{\mathbf{TxtEx}^a}_{\text{strong}}$-complete by Theorem 2. ∎

5 Strong Degrees and Their Characterizations

In this section we establish and characterize a rich structure of degrees of strong reducibility (or, simply, strong degrees), where every degree represents some natural type of learning strategies and reflects topological and algorithmic structures of the languages within it.

Our characterizations of degrees are of two types. Characterizations of the first type, see Theorem 4, specify language classes in and below a given degree. Every such characterization specifies a class of natural strategies learning all languages in the given degree and failing to learn (at least some) languages in the degrees above or incomparable with the given degree. In certain sense, such a characterization establishes the scope of learnability defined by the degree.

Characterizations of the second type, see Theorem 5, specify algorithmic and set-theoretical restrictions on all classes of languages in a given degree and in all degrees above imposed by learnability of hardest classes in the given degree.

Every class $\mathcal{L}$ of languages observed in this paper naturally specifies all classes in the strong degree of this class (that is, all classes that are strongly reducible to the given class, and to which the given class is strongly reducible). We will denote the strong degree of a class $\mathcal{L}$ of languages using the same name as for the class $\mathcal{L}$ itself (for example, *INIT* will stand both for the class $\mathcal{L} = \mathit{INIT}$ defined above, as well as for the whole degree of all classes of languages which are $\equiv^{\mathbf{TxtEx}}_{\text{strong}}$ to *INIT*). Which connotation is being used will be always clear from the context.

The structure of degrees developed in this section can be represented in the form of a complex directed graph. The lowest, or, rather, starting points of our hierarchies, are the degrees *SINGLE*, *COSINGLE*, *INIT* and *COINIT*, that contain well-known classes of languages learnable by some "simplest" strategies. All of these degrees are proven in [JS96] to be pairwise different. A natural class of languages to consider is also *FINITE*. However, this class was shown in [JS96] to be in the same strong degree as *INIT*. The paper [JS96] contains a number of other natural classes of languages, all of which belong to the degrees *SINGLE*, *COSINGLE*, *INIT* or *COINIT*. This enables us to concentrate on classes *SINGLE*, *COSINGLE*, *INIT*, and *COINIT* as the "backbone" of our hierarchy.

Due to space constraints, in this paper, we only concentrate on *INIT*, *COINIT*. Similar characterizations and hierarchy results involving *SINGLE*, *COSINGLE*, in addition to *INIT*, *COINIT* have also been obtained. The reader is referred to [JKW99] for details. Notation and definitions below provide us with terminology and apparatus for these characterizations.

Definition 16 F, a partial recursive mapping from $\mathit{FINITE} \times N$ to N, is called an *up-mapping* iff for all finite sets S, S', for all $j, j' \in N$:

$$\text{If } S \subseteq S' \text{ and } j \leq j', \text{ then}$$
$$F(S,j)\!\downarrow \Rightarrow [F(S',j')\!\downarrow \geq F(S,j)].$$

For an up-mapping F and $L \subseteq N$, we abuse notation slightly and let $F(L)$ denote $\lim_{S \to L, j \to \infty} F(S,j)$ (where by $S \to L$ we mean: take any sequence of finite sets $S_1, S_2, \ldots$, such that $S_i \subseteq S_{i+1}$ and $\bigcup S_i = L$, and then take the limit over these S_i's).

Note that $F(L)$ may be undefined in two ways:

(1) $F(S,j)$ may take arbitrary large values for $S \subseteq L$, and $j \in N$, or

(2) $F(S,j)$ may be undefined for all $S \subseteq L$, $j \in N$.

Definition 17 F, a partial recursive mapping from $\mathit{FINITE} \times N$ to N, is called a *down-mapping* iff for all finite sets S, S' and $j, j' \in N$,

$$\text{If } S \subseteq S' \text{ and } j \leq j', \text{ then}$$
$$F(S,j)\!\downarrow \Rightarrow [F(S',j')\!\downarrow \leq F(S,j)].$$

For a down-mapping F and $L \subseteq N$, we abuse notation slightly and let $F(L) = \lim_{S \to L, j \to \infty} F(S,j)$.

The following results characterize strong degrees below and above *INIT*.

Theorem 4 *$\mathcal{L} \leq^{\mathbf{TxtEx}}_{\text{strong}}$ INIT iff there exist F, a partial recursive up-mapping, and G, a partial limit recursive mapping from N to N, such that for all $L \in \mathcal{L}$,*

(a) $F(L)\!\downarrow < \infty$.

(b) $G(F(L))$ converges to a grammar for L.

Theorem 5 *INIT $\leq^{\mathbf{TxtEx}}_{\text{strong}} \mathcal{L}$ iff there exists a recursive function H such that*

(a) $\{W_{H(i)} \mid i \in N\} \subseteq \mathcal{L}$,

(b) $W_{H(i)} \subset W_{H(i+1)}$, and

(c) $\{W_{H(i)} \mid i \in N\}$ is limiting standardizable.

One can prove similar characterizations as in the above two theorems for *COINIT*, by replacing *INIT* by *COINIT* and in Theorem 4 replacing "up-mapping" by "down-mapping", and in Theorem 5 replacing condition (b) by "$W_{H(i)} \supset W_{H(i+1)}$"; see [JKW99] for details.

The above characterizations describe essential structural and algorithmic properties of the languages in the appropriate degrees.

Every class we have observed represents certain strategies of learning in the limit. Now let us imagine a "multidimensional" language where every "dimension" is being learned using its specific type of learning strategy, that is *SINGLE*, *COSINGLE*, *INIT*, or *COINIT* like. If this idea can be naturally formalized, the following questions can be asked immediately:

1. Are degrees defined by classes of "multidimensional" languages stronger than the degrees of simple "one-dimensional" classes?

2. Is it possible to characterize these degrees in terms similar to the ones we have used for "one-dimensional" degrees?

We consider the following way to form "multidimensional" languages. Therefore, let $\mathit{BASIC} = \{\mathit{INIT}, \mathit{COINIT}\}$. Our approach is based on the following idea: the learner knows in advance to which of the classes from *BASIC* every "dimension" L_k of an "n-dimensional" language L belongs; however, to learn the "dimension" L_{k+1}, one must first learn the codes $i_1, \ldots, i_k$ of the grammars for the languages $L_1, \ldots, L_k$; then L_{k+1} is the $(k+1)$-"projection" $\{x_{k+1} \mid \langle i_1, \ldots, i_k, x_{k+1}, x_{k+2}, \ldots, x_n \rangle \in L\}$.

For example, suppose it is known that the languages L_k (of the k-th "dimension") are from the class *COINIT*. Then, for any L_k, the number i such that $L_k = \{j \mid j \geq i\}$ can be viewed as a legitimate description of this language. Then this $i = i_k$, together with $i_1, i_2, \ldots, i_{k-1}$ found on the previous phases of the learning process and together with some fixed in advance "pattern" (say, *INIT*) (specifying an appropriate learning strategy) can be used to learn the "dimension" L_{k+1}.

"Patterns" specifying classes of languages in different "dimensions" can be of any nature, as long as they provide sufficient information making the class learnable. In our first formalization of this idea below, we limit "patterns" to come from *BASIC*.

Before we give the general definition for the classes that formalizes the above idea, we demonstrate how to define some classes of "two-dimensional" languages based on the classes from *BASIC*. We hope that these definitions and the following discussion will make the general definition and related results more transparent.

Definition 18 $(COINIT, INIT) = \{L \mid$ there exist $i, j \in N$ such that $L = \{\langle a,b\rangle \mid a > i$, or $[a = i$ and $b \leq j]\}\}$.

$(INIT, COINIT) = \{L \mid$ there exist $i, j \in N$ such that $L = \{\langle a,b\rangle \mid a < i$, or $[a = i$ and $b \geq j]\}\}$.

$(INIT, INIT) = \{L \mid$ there exist $i, j \in N$ such that $L = \{\langle a,b\rangle \mid a < i$, or $[a = i$ and $b \leq j]\}\}$.

$(COINIT, COINIT) = \{L \mid$ there exist $i, j \in N$ such that $L = \{\langle a,b\rangle \mid a > i$, or $[a = i$ and $b \geq j]\}\}$.

To justify our definition, we briefly discuss the "natural" strategies that learn the classes defined above.

Consider a language $L \in (COINIT, INIT)$ (see figure 1, where i, j denote the parameters/descriptors of the language L). To learn a language in this class, one first uses a *COINIT*-like strategy, and once the first "descriptor" i of the language has been learned, "changes its mind" to a *INIT*-like strategy to learn the second "descriptor" j. More specifically, imagine the area representing a language in $(COINIT, INIT)$: it consists of the infinite rectangle containing all points $\langle a,b\rangle$ with $a > i$ for some i (apparently, the rectangle is open upword and to the right) and a string of points $\langle i,b\rangle, b \leq j$ just left of the rectangle. The learner first tries to determine the left border i of the rectangle. If some $\langle r,b\rangle$ shows up in the input, $r+1$ can be discarded as a candidate for such i; accordingly, $r+1$ cannot represent the "column" containing the second "dimension" of the language, and, consequently, all pairs $\langle r+1,b\rangle, b \in N$ belong to L, which makes this part of the language easily learnable by *COINIT*-type strategy (only the first "dimension" matters). Once i has been identified (in the limit), the learner, using the "column" $\langle i,\cdot\rangle$, may start to learn the parameter j. Here, if some pair $\langle i,s\rangle$ showed up in the input, $s-1$ can be discarded as a candidate for the parameter j. All discarded pairs $\langle a,b\rangle$ can be viewed as the "terminating" part of the language in question, while $\langle i,j\rangle$ can be viewed as its "propagating" part ("propagating" means "the part of the language representing its description, subject to possible change in the limit").

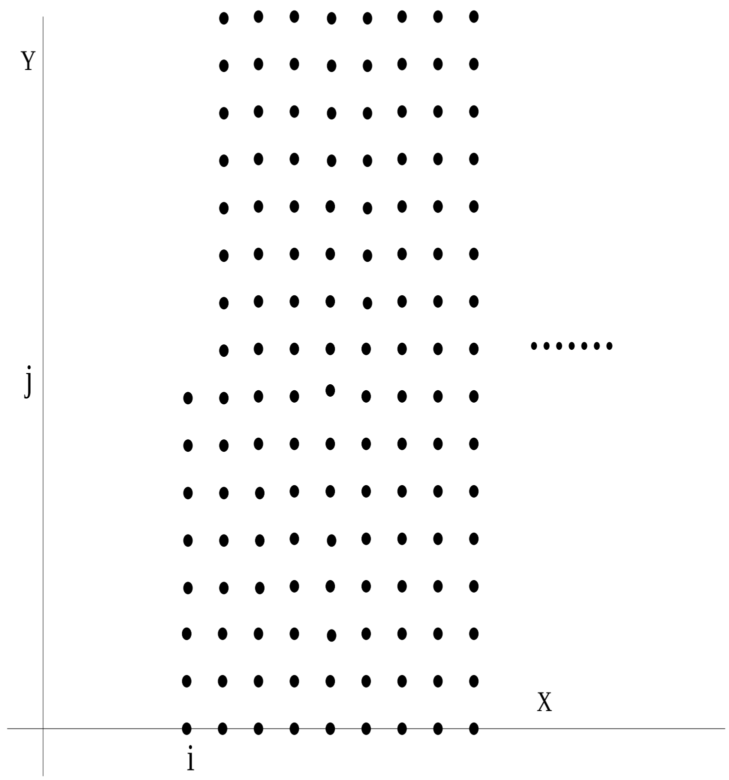

Figure 1: $L_{i,j}^{COINIT,INIT}$

Similar considerations can be applied to $(INIT, COINIT)$, $(INIT, INIT)$, and $(COINIT, COINIT)$.

In some sense, any language L in the above classes consists of two parts:

1. *Terminating* part $T(L)$ consisting of the discarded "conjectures".

2. *Propagating* part $P(L)$ consisting of those pairs in L that represent the current hypothesis-"descriptor" of L.

Now we are ready to give the general definition of "multidimensional" classes formalizing the above approach.

For any tuples X and Y, let $X \cdot Y$ stand for the concatenation of X and Y (that is, $X \cdot Y$ is the tuple, where the first tuple is appended by the components of the second tuple).

Definition 19 Suppose $k \geq 1$. Let $Q \in BASIC^k$. Let $I \in N^k$. Then inductively on k, we define the languages L_I^Q and $T(L_I^Q)$ and $P(L_I^Q)$ as follows.
If $k = 1$, then
(a) if $Q = (INIT)$ and $I = (i)$, then

$T(L_I^Q) = \{\langle x\rangle \mid x < i\}$, $P(L_I^Q) = \{\langle i\rangle\}$, and $L_I^Q = T(L_I^Q) \bigcup P(L_I^Q)$.
(b) if $Q = (COINIT)$ and $I = (i)$, then

$T(L_I^Q) = \{\langle x \rangle \mid x > i\}$, $P(L_I^Q) = \{\langle i \rangle\}$, and $L_I^Q = T(L_i^Q) \bigcup P(L_i^Q)$.

Now suppose we have already defined L_I^Q for $k \le n$. We then define L_I^Q for $k = n+1$ as follows. Suppose $Q = (q_1, \ldots, q_{n+1})$ and $I = (i_1, \ldots, i_{n+1})$. Let $Q_1 = (q_1)$ and $Q_2 = (q_2, \ldots, q_{n+1})$. Let $I_1 = (i_1)$ and $I_2 = (i_2, \ldots, i_{n+1})$. Then,

$T(L_I^Q) = \{X \cdot Y \mid X \in T(L_{I_1}^{Q_1})$, or $[X \in P(L_{I_1}^{Q_1})$ and $Y \in T(L_{I_2}^{Q_2})]\}$,

$P(L_I^Q) = \{X \cdot Y \mid X \in P(L_{I_1}^{Q_1})$ and $Y \in P(L_{I_2}^{Q_2})\}$, and

$L_I^Q = T(L_I^Q) \bigcup P(L_I^Q)$.

For ease of notation we often write $L^Q_{(i_1,i_2,\ldots,i_k)}$ as $L^Q_{i_1,i_2,\ldots,i_k}$.

Definition 20 *Let $Q \in BASIC^k$. Then the class $\mathcal{L}^Q$ is defined as*

$\mathcal{L}^Q = \{L_I^Q \mid I \in N^k\}$.

For technical convenience, for $Q = ()$, $I = ()$, we also define $T(L_I^Q) = \emptyset$, $P(L_Q^I) = \{\langle\rangle\}$, and $L_I^Q = T(L_I^Q) \bigcup P(L_I^Q)$, and $\mathcal{L}^Q = \{L_I^Q\}$.

Note that we have used a slightly different notation for defining the classes $\mathcal{L}^Q$ (for example instead of $(INIT, INIT)$, we now use $\mathcal{L}^{(INIT,INIT)}$). This is for clarity of notation.

One can easily see that the definitions of the "pair"-type classes comply with the general definition. The immediate question is which of the $Q \in BASIC^*$ represent different strong degrees.

We say that a sequence $Q = (q_1, q_2, \ldots, q_k)$ is a subsequence of $Q' = (q'_1, q'_2, \ldots, q'_l)$, iff there exist $i_1, i_2, \ldots, i_k$ such that $1 \le i_1 < i_2 < \ldots < i_k \le l$, and for $1 \le j \le k$, $q_j = q'_{i_j}$.

Theorem 6 (Q-hierarchy Theorem) *Suppose $Q \in BASIC^k$ and $Q' \in BASIC^l$. Then, $\mathcal{L}^Q \le^{\mathbf{TxtEx}}_{\text{strong}} \mathcal{L}^{Q'}$ iff Q is a subsequence of Q'.*

Theorem 6 immediately shows that none of $\mathcal{L}^Q$ is $\le^{\mathbf{TxtEx}}_{\text{strong}}$-complete.

Characterizations for the degrees above and below arbitrary classes $\mathcal{L}^Q$ similar to the ones in Theorems 4 and 5 have been obtained in [JKW99].

The above Q-hierarchy can be applied to quantify intrinsic complexity of learning other classes from texts. Consider, for example, *open semi-hulls* representing the space consisting of all points (x, y) with integer components x, y in the first quadrant of the plane bounded by the y-axis and the "broken" line passing through some points $(0,0), (a_1, c_1), ..., (a_n, c_n)$ with $a_i < a_{i+1}$ (the line is straight between any of the points $(a_i, c_i), (a_{i+1}, c_{i+1})$); further, assume that the slope of the broken line is monotonically non-decreasing (where, for technical convenience, we assume that the first slope is 0: that is $c_1 = 0$). Any such open semi-hull can be easily learned in the limit by the following strategy: given growing finite sets of points in the open semi-hull, learn the first "break" point (a_1, c_1), then the first slope $(c_2 - c_1)/(a_2 - a_1)$, then the second "break" point (a_2, c_2), then the second slope $(c_3 - c_2)/(a_3 - a_2)$, etc. Is this learning strategy optimal? A more general question is: how to measure complexity of learning open semi-hulls? Note that natural complexity measures such as the number of mind changes or memory size would not work, since none of them can be bounded while learning open semi-hulls. One can rather try to determine how many "mind changes" are required in much more general sense: how many times ought a strategy change from *INIT*-like learning to, say, *COINIT*-like learning and back? This is where our hierarchy can be applied. For example, suppose all open semi-hulls with two "angles" are in the class $(INIT, COINIT, INIT, COINIT)$. Then there exists a learning strategy that "changes its mind" from *INIT*-like strategy to *COINIT*, then back to *INIT*, and then one more time to *COINIT* (as a matter of fact, such a strategy for learning the above open semi-hulls exists, and it is somewhat "better" than the natural strategy described above). On the other hand, one can show that no $(COINIT, INIT, COINIT, INIT)$-type strategy (that is, the one that starts like *COINIT*, "changes its mind" to *INIT*, then back to *COINIT*, and then again to *INIT*) can learn open semi-hulls with two "angles". Upper and lower bound of similar kind are obtained for open semi-hulls and other geometrical concepts in [JK99].

In our definition of the classes $\mathcal{L}^Q$ we assumed that the "patterns" for different "dimensions" of a "multidimensional" language come from the set $BASIC$. This gave us opportunity to formalize classes (and degrees) requiring rather complex yet "natural" learning strategies. Now we are going to make another step and define classes of "multidimensional" languages, where such "patterns" come from the *whole* set of vectors Q. Moreover, the grammar for every "dimension" L_k determines which "pattern" Q must be used to learn L_{k+1}.

Note that there exists a recursive bijective mapping, code_k (obtainable effectively in k) from the set of all possible Q (with components from $BASIC$) onto N^k.

Suppose $Q \in BASIC^k$. Let L_i^Q denote the language $L^Q_{i_1,i_2,\cdots,i_k}$, where $i = \langle i_1, \cdots, i_k \rangle$.

Let code be a mapping from $\bigcup_{k=1}^{\infty} BASIC^k$ to N. Let Q^i denote the Q with code i.

Definition 21 *Suppose $S_i = \{i\}$.*

$\mathcal{Q}^0 = \{S_i \mid i \in N\}$.

Let $L^{\mathcal{Q}^m}_{i_0,i_1,\cdots,i_m} = S_{i_0} \times L_{i_1}^{Q^{i_0}} \times \cdots L_{i_m}^{Q^{i_{m-1}}}$.

$\mathcal{Q}^m = \{L^{\mathcal{Q}^m}_{i_0,i_1,\cdots,i_m} \mid i_0, i_1, \cdots, i_m \in N\}$.

We can thus consider $i_0, i_1, \cdots, i_m$ as a parameter of the languages in $\mathcal{Q}^m$.

For example, any language $L \in \mathcal{Q}^1$ consists of all pairs $\langle i, x\rangle$ such that all components x form a language in $\mathcal{L}^{Q^i}$.

Obviously, every class $\mathcal{L}^Q$ is strongly reducible to $\mathcal{Q}^1$. On the other hand, it easily follows from the hierarchy established in Theorem 6 that the degree $\mathcal{Q}^1$ is above any $\mathcal{L}^Q$. It can be shown that $\mathcal{Q}^2 \not\leq^{\mathbf{TxtEx}}_{\text{strong}} \mathcal{Q}^1$. Moreover, it can be shown that all the $\mathcal{Q}^m$ as well as $\mathcal{Q}^* = \bigcup_{m=1}^{\infty} \mathcal{Q}^m$ are not $\leq^{\mathbf{TxtEx}^a}_{\text{strong}}$-complete, for $a \in N \cup \{*\}$.[1] Some open problems are listed in the Conclusions.

6 Weak Degrees and Their Characterizations

First note that for weak-reductions, *INIT*, *FINITE*, *COSINGLE* are $\leq^{\mathbf{TxtEx}}_{\text{weak}}$-complete [JS96]. For all $a \in N \cup \{*\}$, we will give a characterization of $\leq^{\mathbf{TxtEx}^a}_{\text{weak}}$-complete classes below. A characterization of degrees involving *COINIT* and *SINGLE*, as well as a hierarchy based on the classes $\mathcal{L}^Q$, where $Q \in \{SINGLE, COINIT\}^*$, has also been obtained. The reader is referred to [JKW99] for details.

Definition 22 A non-empty class $\mathcal{L}$ of languages is called *quasi-dense* iff

(a) $\mathcal{L}$ is 1–1 recursively enumerable.

(b) For any $L \in \mathcal{L}$ and any finite $S \subseteq L$, there exists an $L' \in \mathcal{L}$, such that $S \subseteq L'$, but $L \neq L'$.

Note: (b) can be equivalently replaced by

(b') For any finite set S, either there exists no language in $\mathcal{L}$ extending S, or there exist infinitely many languages in $\mathcal{L}$ extending S.

Theorem 7 *For any $a \in N \cup \{*\}$ and any $\mathcal{L} \in \mathbf{TxtEx}^a$, $\mathcal{L}$ is $\leq^{\mathbf{TxtEx}^a}_{\text{weak}}$-complete iff there exists a quasi-dense subclass of $\mathcal{L}$ which is a-limiting standardizable.*

7 Conclusions

The formalisms and results obtained in the paper are of two types:

a) Formalisms, hierarchies, and characterizations for classes of "multidimensional" languages, where information learned from one "dimension" aids to learn another one. The characterizations define set-theoretical and algorithmic properties of such classes. The obtained hierarchies, as has been demonstrated in [JK99] in more detail, can be used as scales for quantifying complexity of learning other classes of languages.

[1]For the definition of $\mathcal{Q}^*$ we assume that there is some uniform way in which one can determine the size of the tuples, for example by coding any tuple x in N^k, as $\langle k, x\rangle$.

b) The characterizations of complete degrees. These characterizations specify algorithmic and topological properties of classes in the complete degrees. A new natural powerful class of languages complete for strong reductions has been discovered.

The results for "multidimensional" languages reveal a new variety of learning strategies, which, to learn a "dimension", use previously learned information to find the right "subspace", or a previously learned "pattern" specifying a learning "substrategy" for the next "dimension". As far as the former approach is concerned, the picture of hierarchies based on "core" classes *SINGLE*, *COSINGLE*, *INIT*, *COINIT* (*SINGLE*, *COINIT* for weak reductions) has been completed. The latter approach is implemented in the form of classes $\mathcal{Q}^m$ and $\mathcal{Q}^*$, see Definition 21. There is a number of interesting open problems related to these classes, as well as to the formalism as a whole:

a) Do the classes $\mathcal{Q}^m$ for $m > 1$ form an infinite hierarchy?

b) Is it possible to define a "natural" class of languages based on combinations of classes from *BASIC* *above* the class $\mathcal{Q}^*$?

c) Is it possible to (naturally) define a type of language classes with a different way of using or learning "patterns"?

The degrees of "core" classes forming *BASIC* are known to contain many of important "practical" learning problems. For example, *COINIT* contains the class of *pattern languages* [JS96]. However, there certainly exist "natural" classes of infinite/finite languages that are probably incomparable, at least in terms of strong reductions, with some/all classes in *BASIC*. One can add these classes to *BASIC* and apply the formalisms developed in the paper. Exploration of, say, *Q*-classes based on such extensions of *BASIC* can give a deeper understanding of the nature of learning strategies and learning from texts as a whole.

8 Acknowledgements

Sanjay Jain was supported in part by NUS grant number RP3992710. Efim Kinber was supported in part by the URCG grant of Sacred Heart University.

References

[BB75] L. Blum and M. Blum. Toward a mathematical theory of inductive inference. *Information and Control*, 28:125–155, 1975.

[BF72] J. Bārzdiņš and R. Freivalds. On the prediction of general recursive functions. *Soviet Mathematics Doklady*, 13:1224–1228, 1972.

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

[Cas88] J. Case. The power of vacillation. In D. Haussler and L. Pitt, editors, *Proceedings* *of the Workshop on Computational Learning Theory*, pages 133–142. Morgan Kaufmann, 1988.

[CL82] J. Case and C. Lynes. Machine inductive inference and language identification. In M. Nielsen and E. M. Schmidt, editors, *Proceedings of the 9th International Colloquium on Automata, Languages and Programming*, volume 140 of *Lecture Notes in Computer Science*, pages 107–115. Springer-Verlag, 1982.

[CS83] J. Case and C. Smith. Comparison of identification criteria for machine inductive inference. *Theoretical Computer Science*, 25:193–220, 1983.

[DS86] R. Daley and C. Smith. On the complexity of inductive inference. *Information and Control*, 69:12–40, 1986.

[Fel72] J. Feldman. Some decidability results on grammatical inference and complexity. *Information and Control*, 20:244–262, 1972.

[FKS95] R. Freivalds, E. Kinber, and C. Smith. On the intrinsic complexity of learning. *Information and Computation*, 123(1):64–71, 1995.

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

[Ful90] M. Fulk. Prudence and other conditions on formal language learning. *Information and Computation*, 85:1–11, 1990.

[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, 1979.

[JK99] S. Jain and E. Kinber. On intrinsic complexity of learning geometrical concepts from texts. Tech. Rep. TRB6/99, School of Computing, National University of Singapore, 1999.

[JKW99] S. Jain, E. Kinber, and R. Wiehagen. Language learning: Degrees of intrinsic complexity and their characterizations. Tech. Rep. LSA-99-02E, Centre for Learning Systems and Applications, Department of Computer Science, University of Kaiserslautern, Germany, 1999.

[JORS99] S. Jain, D. Osherson, J. Royer, and A. Sharma. *Systems that Learn: An Introduction to Learning Theory.* MIT Press, Cambridge, Mass., second edition, 1999.

[JS94] S. Jain and A. Sharma. Characterizing language learning by standardizing operations. *Journal of Computer and System Sciences*, 49(1):96–107, 1994.

[JS96] S. Jain and A. Sharma. The intrinsic complexity of language identification. *Journal of Computer and System Sciences*, 52:393–402, 1996.

[JS97] S. Jain and A. Sharma. The structure of intrinsic complexity of learning. *Journal of Symbolic Logic*, 62:1187–1201, 1997.

[Kin75] E. Kinber. On comparison of limit identification and limit standardization of general recursive functions. *Uch. zap. Latv. univ.*, 233:45–56, 1975.

[Kin94] E. Kinber. Monotonicity versus efficiency for learning languages from texts. In S. Arikawa and K. Jantke, editors, *Algorithmic Learning Theory: Fourth International Workshop on Analogical and Inductive Inference (AII '94) and Fifth International Workshop on Algorithmic Learning Theory (ALT '94)*, volume 872 of *Lecture Notes in Artificial Intelligence*, pages 395–406. Springer-Verlag, 1994.

[KPSW99] E. Kinber, C. Papazian, C. Smith, and R. Wiehagen. On the intrinsic complexity of learning recursive functions. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, pages 257–266. ACM Press, 1999.

[KS95] E. Kinber and F. Stephan. Language learning from texts: Mind changes, limited memory and monotonicity. *Information and Computation*, 123:224–241, 1995.

[LZ93] S. Lange and T. Zeugmann. Learning recursive languages with a bounded number of mind changes. *International Journal of Foundations of Computer Science*, 4(2):157–178, 1993.

[MY78] M. Machtey and P. Young. *An Introduction to the General Theory of Algorithms.* North Holland, New York, 1978.

[OW82] D. Osherson and S. Weinstein. Criteria of language learning. *Information and Control*, 52:123–138, 1982.

[Rog67] H. Rogers. *Theory of Recursive Functions and Effective Computability.* McGraw-Hill, 1967. Reprinted, MIT Press 1987.

[Wie86] R. Wiehagen. On the complexity of program synthesis from examples. *Journal of Information Processing and Cybernetics (EIK)*, 22:305–323, 1986.