

Characterizing Language Identification by Standardizing Operations

Sanjay Jain

Department of Computer and Information Sciences
University of Delaware
Newark, Delaware 19716
USA
Email: sjain@cis.udel.edu

Arun Sharma

School of Computer Science and Engineering
The University of New South Wales
Sydney, NSW, 2033
Australia
Email: arun@spectrum.cs.unsw.oz.au

Abstract

Notions from formal language learning theory are characterized in terms of standardizing operations on classes of recursively enumerable languages. Algorithmic *identification* in the limit of grammars from *text* presentation of recursively enumerable languages is a central paradigm of language learning. A mapping, \mathbf{F} , from the set of all grammars into the set of all grammars is a *standardizing operation* on a class of recursively enumerable languages \mathcal{L} just in case \mathbf{F} maps any grammar for any language $L \in \mathcal{L}$ to a *canonical* grammar for L . Investigating connections between these two notions is the subject of this paper.

1 Introduction

A child (modeled as a machine) receives (in arbitrary order) all the well-defined strings of a language (a *text* for the language) L , and simultaneously conjectures a succession of candidate grammars for the language being received. A criterion of success is for the child to eventually conjecture a correct grammar for L and to never change its conjecture thereafter. If, in this scenario for success, the child machine is replaced by an algorithmic machine \mathbf{M} , then we say that \mathbf{M} **TextEx**-identifies L . **TextEx** is defined to be the class of sets \mathcal{L} of recursively enumerable languages such that some machine **TextEx**-identifies each member of \mathcal{L} .

TextEx-identification is essentially Gold's [10] seminal notion of *identification*. The reader is directed to [18, 23, 22, 16] for a discussion of the influence of this paradigm on contemporary theories of natural language. The present paper studies characterizations of the class **TextEx** and some of its extensions. The usefulness of this study is apparent as similar characterizations have been used by Freivalds [5], Chen [4], and Case, Jain, and Sharma [3] to study program size restrictions in computational learning. Our study is motivated by analogous work of Freivalds, Kinber, and Wiehagen [6] in the context of algorithmic inference of programs from graphs of recursive functions.

We now motivate the notion of a *standardizing operation* on a class of recursively enumerable (r.e.) languages. \mathcal{L} , a class of languages, is *effectively standardizable* just in case there exists a partial recursive function p such that for all $L \in \mathcal{L}$, for all i and j such that i and j are grammars for L , $p(i)\downarrow = p(j)\downarrow$ and $p(i)$ is a grammar for L . We say that a recursive function r with two arguments defines a *limit-effective language operation* on a class of recursively enumerable languages \mathcal{L} just in case for any language L in \mathcal{L} and any grammar i for L , $\lim_{n \rightarrow \infty} r(i, n)$ exists and is the same for all grammars of L . If, in the definition of limit-effective language operation, the limiting value also happens to be a grammar for L , then r defines a

standardizing limit-effective language operation on \mathcal{L} . We say that \mathcal{L} is *limit-effective language standardizable* just in case there exists a *standardizing limit-effective language operation* on \mathcal{L} . **Lels** is defined to be the class of sets \mathcal{L} of recursively enumerable languages such that \mathcal{L} is limit-effective language standardizable.

It is shown that **TxtEx** is properly contained in **Lels**. To characterize **TxtEx** in terms of standardizing operations, we define a restricted form of standardizing limit-effective language operation (viz., *continuously limit-effective language standardizable operation*), such that the corresponding class (viz., **Cels**) is exactly equal to **TxtEx**.

Since, **TxtEx** is properly contained in **Lels**, to get a characterization of **Lels** in terms of language learning notions, we borrow extensions on the theme of **TxtEx**-identification from [13]. We require that a learning machine, trying to infer a grammar for a language from its *text*, be presented with an *upper-bound* on the minimal grammar for the language being learned. This is plausible additional information, as an upper-bound on the size of “human brain storage” can be thought of as an upper-bound on the size of a grammar for any language that can be learned by a child. This generalization of Gold’s notion gives us a new criterion for language learning. A machine is said to **TxtBex**-*identify* a language L just in case the machine, when fed any *text* for L and an upper-bound on the minimal grammar for L , converges to a correct grammar for L (**B** stands for “**B**ound”). **TxtBex** is defined to be the class of sets \mathcal{L} of recursively enumerable languages such that some machine **TxtBex**-identifies each language in \mathcal{L} . In the definition of **TxtBex**-identification if we further require that the machine infer the same grammar for any upper-bound, we get a new criteria of language learning called **TxtUniBex**-*identification* (**Uni** stands for “**U**nique”). The class **TxtUniBex** can be similarly defined. We show that the class **TxtUniBex** is exactly equal to **Lels**, and **Lels** is properly contained in **TxtBex**.

2 Notation

Recursion-theoretic concepts not explained below are treated in [20]. N is the set of natural numbers. $a, b, c, i, j, k, l, m, n, x$, and y , with or without decorations (decorations are subscripts, superscripts and the like), range over natural numbers unless otherwise specified. $\subseteq, \subset, \supseteq, \supset$, denote subset, proper subset, superset and proper superset respectively. \in denotes ‘element of.’ N_a denotes the set $\{x \in N \mid x \leq a\}$. \emptyset denotes the empty set. S , with or without decorations, ranges over subsets of N . D_x denotes the finite set whose canonical index is x [20]. According to Rogers’ scheme, $D_0 = \emptyset$. $\text{card}(S)$ denotes the cardinality of the set S . $\max(\cdot), \min(\cdot)$ denote the maximum and minimum of a set, respectively. By convention $\max(\emptyset) = 0$ and $\min(\emptyset) = \infty$. $\mu x[Q(x)]$ is the least integer x such that the predicate $Q(x)$ is true, if such a least integer exists; $\mu x[Q(x)]$ is undefined otherwise. For any set A , 2^A denotes the power set of A .

p, q range over partial recursive functions. f, g, r, s range over total recursive functions. The set of all total recursive functions of one variable is denoted by \mathcal{R} . For $n > 0$, \mathcal{R}^n denotes the set of total recursive functions of n variables. For a partial recursive function p , $\text{domain}(p)$ denotes the domain of p and $\text{range}(p)$ denotes the range of p . \downarrow denotes defined. \uparrow denotes undefined. $p(x) \downarrow$ iff $x \in \text{domain}(p)$; $p(x) \uparrow$ otherwise.

L denotes a recursively enumerable (r.e.) subset of N (also referred to as an r.e. language). \mathcal{E} denotes the class of all r.e. languages. \mathcal{L} , with or without decorations, ranges over subsets of \mathcal{E} . φ denotes a standard acceptable programming system (also referred to as standard acceptable numbering) [19, 20]. φ_i denotes the partial recursive function computed by the i^{th} program in the standard acceptable programming system φ . We often refer to the i^{th} program in the φ system as φ -program i . $\text{MinProg}(f)$ denotes the minimal program for f in the φ programming system. W_i denotes the domain of φ_i . W_i is, then, the r.e. set/language ($\subseteq N$) accepted by φ -program i . We can (and do) also think of i as (coding) a (type 0 [11])

grammar for generating W_i . $\text{MinGram}(L)$ denotes the minimal grammar for L in the φ programming system. Φ denotes an arbitrary Blum complexity measure [2] for φ . $W_{i,n}$ denotes the set $\{x < n \mid \Phi_i(x) < n\}$.

$\langle i, j \rangle$ stands for an arbitrary computable one to one encoding of all pairs of natural numbers onto N [20]. Corresponding projection functions are π_1 and π_2 . $(\forall i, j \in N) [\pi_1(\langle i, j \rangle) = i \text{ and } \pi_2(\langle i, j \rangle) = j \text{ and } \langle \pi_1(x), \pi_2(x) \rangle = x]$. Similarly, $\langle i_1, i_2, \dots, i_n \rangle$ denotes a computable one to one encoding of all n -tuples onto N . It should be noted that we will sometimes abuse the notation slightly and refer to $\langle x, y \rangle$ as $\langle D_x, y \rangle$, i.e., we will write the name of the finite set in the first argument instead of its canonical index. This is for simplicity of presentation and it will be clear when we resort to such an interpretation.

The quantifiers ' $\overset{\infty}{\forall}$ ' and ' $\overset{\infty}{\exists}$ ' mean 'for all but finitely many' and 'there exists infinitely many', respectively.

3 Preliminaries

In this section, we briefly describe notions and results from the recursion theoretic machine learning literature. We first introduce a notion that facilitates discussion about elements of a language being fed to a learning machine.

A *finite sequence* is a mapping from $\{x \mid x < a\}$, for some $a \in N$, into $(N \cup \{\#\})$. We let σ and τ , with or without decorations, range over finite sequences. The *content* of a finite sequence σ , denoted $\text{content}(\sigma)$, is the set of natural numbers in the range of σ . Intuitively, $\#$'s represent pauses in the presentation of data. The *length* of σ , denoted $|\sigma|$, is the number of elements in the domain of σ . $\sigma \subset \tau$ means that σ is an initial sequence of τ . SEQ denotes the set of all finite sequences.

Definition 1 A *learning machine* is an algorithmic device which computes a mapping from SEQ into N .

We let \mathbf{M} , with or without decorations, range over learning machines.

Definition 2 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 .

We let T , with or without decorations, range over texts. $T[n]$ denotes the finite initial sequence of T with length n . Hence, $\text{domain}(T[n]) = \{x \mid x < n\}$. Suppose \mathbf{M} is a learning machine and T is a text. $\mathbf{M}(T)\downarrow$ (read: $\mathbf{M}(T)$ *converges*) $\iff (\exists i)(\forall n)[\mathbf{M}(T[n]) = i]$. If $\mathbf{M}(T)\downarrow$, then $\mathbf{M}(T)$ is defined = the unique i such that $(\forall n)[\mathbf{M}(T[n]) = i]$; otherwise we say that $\mathbf{M}(T)$ *diverges* (written: $\mathbf{M}(T)\uparrow$). Convergence of \mathbf{M} on T is also referred to as convergence *in the limit*.

Definition 3 [10]

- (a) \mathbf{M} **TxtEx**-*identifies* L (written: $L \in \mathbf{TxtEx}(\mathbf{M})$) just in case $(\forall \text{ texts } T \text{ for } L)[\mathbf{M}(T)\downarrow \wedge W_{\mathbf{M}(T)} = L]$.
- (b) $\mathbf{TxtEx} = \{\mathcal{L} \subseteq \mathcal{E} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{TxtEx}(\mathbf{M})]\}$.

Below, we define certain restrictions on learning machines and state results describing the effects of these restrictions.

Definition 4

- (a) [1] A learning machine \mathbf{M} is *order-independent* just in case for every $L \in \mathbf{TxtEx}(\mathbf{M})$ and for every pair of texts T and T' for L , $\mathbf{M}(T) = \mathbf{M}(T')$.
- (b) [8, 21] A learning machine \mathbf{M} is *rearrangement-independent* just in case $(\forall \sigma_1, \sigma_2) [[\text{content}(\sigma_1) = \text{content}(\sigma_2) \wedge |\sigma_1| = |\sigma_2|] \Rightarrow \mathbf{M}(\sigma_1) = \mathbf{M}(\sigma_2)]$.

Lemma 5 [8, 21] *From any learning machine \mathbf{M} one may effectively construct \mathbf{M}' such that (1) through (3) all hold.*

- (1) $\mathbf{TxtEx}(\mathbf{M}) \subseteq \mathbf{TxtEx}(\mathbf{M}')$.
- (2) \mathbf{M}' is *order-independent*.
- (3) \mathbf{M}' is *rearrangement-independent*.

We now introduce a technical result, Lemma 7, due to L. Blum and M. Blum. This result is helpful in the description of one of our results.

Definition 6

- (a) [8] σ is a **TxtEx**-stabilizing sequence for \mathbf{M} on L just in case $\text{content}(\sigma) \subseteq L$ and $(\forall \sigma' \mid \text{content}(\sigma') \subseteq L \wedge \sigma \subseteq \sigma')[\mathbf{M}(\sigma') = \mathbf{M}(\sigma)]$.
- (b) [1, 17] σ is a **TxtEx**-locking sequence for \mathbf{M} on L just in case σ is a **TxtEx**-stabilizing sequence for \mathbf{M} on L and $W_{\mathbf{M}(\sigma)} = L$.

Lemma 7 [1, 17] *If \mathbf{M} **TxtEx**-identifies L , then there is a **TxtEx**-locking sequence for \mathbf{M} on L .*

If $\mathcal{L} \in \mathbf{TxtEx}$, then, using Lemma 5, we can say, without loss of generality, that \mathcal{L} is **TxtEx**-identified by a rearrangement-independent and order-independent machine \mathbf{M}' . Lemma 7 states that if \mathbf{M} **TxtEx**-identifies L , then there is a **TxtEx**-locking sequence for \mathbf{M} on L . If \mathbf{M} is rearrangement-independent, then output of \mathbf{M} , on input σ , is completely determined by $\text{content}(\sigma)$ and $|\sigma|$. Hence, when we are considering machines which are rearrangement-independent we will frequently refer to a finite sequence σ by $\langle x, l \rangle$ where $D_x = \text{content}(\sigma)$ and $l = |\sigma|$. For a given rearrangement-independent machine \mathbf{M} and a language L , the least number $\langle x, l \rangle$, such that $\langle x, l \rangle$ is a **TxtEx**-locking sequence for \mathbf{M} on L is called *the least **TxtEx**-locking sequence for \mathbf{M} on L* . For ease of discussion, we will abuse the notation slightly, and often refer to $\langle x, l \rangle$ by $\langle D_x, l \rangle$.

4 **TxtEx**-Identification with Additional Information

It could be argued that a language learner makes use of, in addition to a text presentation, some additional information about the language. An upper-bound on the size of the minimal grammar of the language being learned is one such possible additional information. In the present section, we briefly

consider the resulting learning criteria with additional information. One of these criteria turns out to be equivalent to a notion introduced in the present paper. It is technically expedient to treat our learning machines to act on two arguments: additional information and finite sequence. It will be clear from the context if we are discussing learning with additional information or learning without additional information.

$\mathbf{M}(b, \sigma)$ denotes the output of \mathbf{M} on additional information b and a finite sequence σ . For the criteria of inference discussed in this paper we can and do assume, without loss of generality, that $\mathbf{M}(b, \sigma)$ is always defined. $\mathbf{M}(b, T)\downarrow = i \iff (\forall n)[\mathbf{M}(b, T[n]) = i]$. We write $\mathbf{M}(b, T)\downarrow \iff (\exists i)[\mathbf{M}(b, T)\downarrow = i]$.

Definition 8 [13]

- (a) \mathbf{M} **TxtBex**-identifies L (written: $L \in \mathbf{TxtBex}(\mathbf{M})$) $\iff (\forall b \geq \text{MinGram}(L)) (\forall T \text{ for } L) (\exists i \mid W_i = L)[\mathbf{M}(b, T)\downarrow = i]$.
- (b) $\mathbf{TxtBex} = \{\mathcal{L} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{TxtBex}(\mathbf{M})]\}$.

Intuitively, a language learning machine \mathbf{M} **TxtBex**-identifies a language L just in case \mathbf{M} , presented with any b at least as large as the minimal grammar for L and any text for L , converges in the limit to a grammar for L . If we further require that the grammar inferred in the limit be the same for any upper-bound, we get a new language learning criteria described below.

Definition 9 [13]

- (a) \mathbf{M} **TxtUniBex**-identifies L (written: $L \in \mathbf{TxtUniBex}(\mathbf{M})$) $\iff (\exists i \mid W_i = L) (\forall b \geq \text{MinGram}(L)) (\forall T \text{ for } L) [\mathbf{M}(b, T)\downarrow = i]$.
- (b) $\mathbf{TxtUniBex} = \{\mathcal{L} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{TxtUniBex}(\mathbf{M})]\}$.

Intuitively, a learning machine \mathbf{M} **TxtUniBex**-identifies L just in case \mathbf{M} infers in the limit a unique grammar for L upon being fed any upper-bound for the minimal grammar of L and any text for L .

For the purposes of the present paper, the above definitions suffice; the reader is directed to [13] for an extensive study of the classes **TxtUniBex**,

TxtBex, and their generalizations. Fulk [8], and Jain and Sharma [12] provide other approaches to modeling additional information for a language learning agent (also see [7]). We now state the relationship between the classes **TxtEx**, **TxtUniBex**, and **TxtBex**. Theorems 10 and 11 below can also be derived using results by Kinber [14] cited in [6].

Theorem 10 [13] $\mathbf{TxtEx} \subset \mathbf{TxtUniBex}$.

Theorem 11 [13] $\mathbf{TxtUniBex} \subset \mathbf{TxtBex}$.

Theorem 12 [13] $\mathcal{E} \not\subset \mathbf{TxtBex}$.

We summarize the relationship between various classes defined in this section.

$$\mathbf{TxtEx} \subset \mathbf{TxtUniBex} \subset \mathbf{TxtBex} \subset 2^{\mathcal{E}}.$$

In the next section, we show our main results which provide a characterization for the classes **TxtEx** and **TxtUniBex** in terms of standardizing operations.

5 Connections between Language Learning and Standardizing Operations

We now characterize **TxtEx** and **TxtUniBex** in terms of standardizing operations on classes of r.e. languages. To this end, we first formally define the notion of a limit-effective language operation on a set of r.e. languages.

Definition 13 \mathbf{F} , a mapping from \mathcal{E} to N , is a *limit-effective language operation* for $\mathcal{L} \iff (\exists r \in \mathcal{R}^2) [(\forall L \in \mathcal{L}) (\forall j) [(W_j = L) \Rightarrow \lim_{n \rightarrow \infty} r(j, n) = \mathbf{F}(L)]]$. We say that r defines the *limit-effective language operation* \mathbf{F} on \mathcal{L} . For $L \in \mathcal{L}$, we denote $\mathbf{F}(L)$ by r_L .

Intuitively, a limit-effective language operation on a class of r.e. languages \mathcal{L} behaves thus: given any grammar for a language $L \in \mathcal{L}$, it finds (in the limit) a *unique* number for that L . Additionally, if the unique number also happens to be a grammar for L , then we refer to such a limit-effective language operation as a *standardizing* limit-effective language operation. This is the subject of next definition.

Definition 14

- (a) \mathbf{F} , a mapping from \mathcal{E} to N , is a *standardizing limit-effective language operation* for $\mathcal{L} \iff [\mathbf{F} \text{ is a } \textit{limit-effective language operation} \text{ on } \mathcal{L}] \wedge (\forall L \in \mathcal{L}) [W_{\mathbf{F}(L)} = L]$.
- (b) \mathcal{L} is *limit-effective language standardizable* just in case there exists a *standardizing limit-effective language operation* \mathbf{F} on \mathcal{L} .
- (c) $\mathbf{Lels} = \{ \mathcal{L} \subseteq \mathcal{E} \mid \mathcal{L} \text{ is } \textit{limit-effective language standardizable} \}$.

If $s \in \mathcal{R}^2$ defines a limit-effective language operation \mathbf{F} on \mathcal{L} and \mathbf{F} is also a standardizing limit-effective language operation on \mathcal{L} , then we say that s defines the standardizing limit-effective language operation \mathbf{F} on \mathcal{L} . In this case we denote $F(L)$ by s_L .

We give some intuitive insight into the notion of \mathcal{L} being limit-effective language standardizable. The interpretation below was pointed out to us by John Case. The *grammar equivalence problem* ($\{ \langle x, y \rangle \mid W_x = W_y \}$) is well-known to be Π_2^0 -complete [20]; hence, it cannot be *accepted* by a *limiting* recursive procedure. The role of \mathbf{F} in the definition of limit-effective language standardizable is to indirectly provide a limiting recursive solution to this problem for the special case where the grammars generate languages in \mathcal{L} : \mathbf{F} finds (in the limit) *canonical* grammars.

\mathbf{Lels} is a collection of all limit-effective language standardizable classes of r.e. languages. Theorem 15 below shows that \mathbf{Lels} is exactly the class of r.e. languages that can be **TxtUniBex**-identified.

Theorem 15 **TxtUniBex** = **Lels**.

Proof: Let $\mathcal{L} \in \mathbf{TxtUniBex}$. We show that $\mathcal{L} \in \mathbf{Lels}$. Let \mathbf{M} **TxtUniBex**-identify \mathcal{L} . We define a limit-effective language operation s that witnesses $\mathcal{L} \in \mathbf{Lels}$. Let σ_j^n uniformly denote a finite sequence such that $\sigma_j^n \subset \sigma_j^{n+1}$ and $\text{content}(\sigma_j^n) = W_{j,n}$. Let $s(j, n) = \mathbf{M}(j, \sigma_j^n)$. For any $L \in \mathcal{L}$, let a_L be such that \mathbf{M} , on any text for L and any $b \geq \text{MinGram}(L)$, converges to a_L . Then, clearly $\lim_{n \rightarrow \infty} s(j, n) = a_L$. Thus, $\mathcal{L} \in \mathbf{Lels}$. This shows $\mathbf{TxtUniBex} \subseteq \mathbf{Lels}$.

We now show that $\mathbf{Lels} \subseteq \mathbf{TxtUniBex}$. Let $\mathcal{L} \in \mathbf{Lels}$. Let s define a standardizing limit-effective language operation witnessing $\mathcal{L} \in \mathbf{Lels}$. We now give the construction for a language learning machine \mathbf{M} that **TxtUniBex**-identifies \mathcal{L} .

```

begin { $\mathbf{M}(b, T[n])$ }
  1. Let  $a_0 = \max(\{a \mid (a \leq n) \wedge (\exists j \leq b) [W_{j,a} \subseteq \text{content}(T[n]) \wedge W_{j,n} \supseteq \text{content}(T[a])]\})$ .
  2. Let  $j_0 = \min(\{j \mid (j \leq b) \wedge [W_{j,a_0} \subseteq \text{content}(T[n]) \wedge W_{j,n} \supseteq \text{content}(T[a_0])]\})$ .
  3. Output  $s(j_0, n)$ .
end

```

Now we show that \mathbf{M} **TxtUniBex**-identifies \mathcal{L} .

For any $L \in \mathcal{L}$ and any $b \geq \text{MinGram}(L)$, let $S = \{j \mid j \leq b \wedge W_j = L\}$.

For any text T for L , let n_0, n_1 be so large that the following hold:

- 1) $(\forall i \in (N_b - S))[W_i \not\supseteq \text{content}(T[n_0]) \vee W_{i,n_0} \not\subseteq L]$; and
- 2) $(\forall j \in S)(\forall n \geq n_1)[s(j, n) = s_L \wedge W_{j,n_1} \supseteq \text{content}(T[n_0 + 1]) \wedge W_{j,n_0+1} \subseteq \text{content}(T[n_1])]$.

Clearly, such n_0, n_1 exist. Now, $(\forall n \geq \max(\{n_0, n_1\}))$, \mathbf{M} , on input b and $T[n]$, outputs $s(j, n) = s_L$ for some $j \in S$. Hence, \mathbf{M} **TxtUniBex**-identifies \mathcal{L} . ■

Our main aim is to characterize **TxtEx** in terms of limit-effective language operations. But, the above result tells us that the notion of limit-effective language standardizable class is too general, and hence we need

to come up with a more restricted notion. We do exactly this by defining a *continuously limit-effective language standardizable* class in Definition 17. But, first we introduce the following useful technical concept.

Definition 16 Let $a \in N$. A finite set D is said to be *a-consistent* with an r.e. language $L \iff [[D \subseteq L] \wedge [(D \cap N_a) = (L \cap N_a)]]$.

Intuitively, $D \subseteq L$ is *a-consistent* with L just in case for each $i \leq a$, $i \in D \iff i \in L$.

Definition 17

(a) \mathcal{L} is *continuously limit-effective language standardizable* $\iff (\exists r, s \in \mathcal{R}^2)$ such that the following hold:

1. r defines a limit-effective language operation on \mathcal{L} ;
2. s defines a standardizing limit-effective language operation on \mathcal{L} ;
3. $(\forall L \in \mathcal{L})$
 - 3a. $[D_{r_L}$ is $\max(D_{r_L})$ -consistent with $L]$ and
 - 3b. $(\exists l_L \in N) (\forall n \geq l_L) (\forall j) [[D_{r(j,n)}$ is $\max(D_{r_L})$ -consistent with $L] \Rightarrow [s(j, n) = s_L]]$.

(b) $\mathbf{Clels} = \{\mathcal{L} \subseteq \mathcal{E} \mid \mathcal{L} \text{ is continuously limit-effective language standardizable}\}$.

s , in the above definition of a continuously limit-effective language standardizable class has the same role as \mathbf{F} in the definition of a limit-effective language standardizable class. r , another limit-effective language operation, places some extra constraints on how s finds (in the limit) *canonical* grammars for languages in \mathcal{L} .

Theorem 18 $\mathbf{TxtEx} = \mathbf{Clels}$.

Proof: Let $\mathcal{L} \in \mathbf{TxtEx}$. We show that $\mathcal{L} \in \mathbf{Clels}$. Let \mathbf{M} \mathbf{TxtEx} -identify \mathcal{L} . Without loss of generality, let \mathbf{M} be rearrangement independent and order independent.

begin $\{r(j, n)\}$

1. {We search for the least locking sequence of \mathbf{M} on W_j }
 find the least $\langle D, l \rangle$ such that $(D \subseteq W_{j,n})$ and
 $(\forall S \mid D \subseteq S \subseteq W_{j,n}) (\forall l' \mid (\text{card}(S) - \text{card}(D) + l) \leq l' \leq n)$
 $[\mathbf{M}(\langle D, l \rangle) = \mathbf{M}(\langle S, l' \rangle)]$.
 {Clearly, such a $\langle D, l \rangle$ exists, since, for $D = W_{j,n}$ and $l = n$, the
 above is vacuously true.}
 Let $a = \max(D)$. Let $D' = \{x \mid x \in W_{j,n} \cap N_a\}$.
2. **if** $(\forall i \leq n) [\mathbf{M}(\langle D, l \rangle) = \mathbf{M}(\langle D', l + \text{card}(D') + i \rangle)]$ **then**
 define $r(j, n) = k$ such that $D_k = D'$
else
 define $r(j, n) = 0$
endif

end

Let i_0 be a grammar for the empty set.

begin $\{s(j, n)\}$

- if** $r(j, n) = 0$ **then**
 let $s(j, n) = i_0$
 {note that according to our convention $D_0 = \emptyset$ }
- else**
 let $s(j, n) = \mathbf{M}(\langle D, l \rangle)$, where D, l are as found in step 1 of
 the definition of $r(j, n)$.

endif

end

Claim 19 r defines a limit-effective language operation for \mathcal{L} .

Proof: Clearly, r is a total recursive function. If $W_i = W_j = L \in \mathcal{L}$, then for large enough n , D and l as found in the procedure for $r(i, n)$ and $r(j, n)$ will be such that $\langle D, l \rangle$ is the least **TextEx** locking sequence for \mathbf{M} on L . Hence, for large enough n , D' found in step 1 of the procedure for $r(i, n)$

and $r(j, n)$ would also be the same. Thus, $\lim_{n \rightarrow \infty} r(i, n) = \lim_{n \rightarrow \infty} r(j, n)$.

■

Claim 20 *s defines a standardizing limit-effective language operation for \mathcal{L} .*

Proof: Arguing as in Claim 19, we can show that s is a limit-effective language operation for \mathcal{L} . Also, for all j such that $W_j \in \mathcal{L}$, for large enough n , $\langle D, l \rangle$ as found in step 1 of the procedure for $r(j, n)$ is the least locking sequence for \mathbf{M} on L (since, \mathbf{M} **TextEx**-identifies W_j) and, thus, we have that $W_{s_L} = L$. ■

We now define l_L for each $L \in \mathcal{L}$. For all $L \in \mathcal{L}$, let $l_L = l$ and $S_L = D$, where $\langle D, l \rangle$ is the least **TextEx** locking sequence for \mathbf{M} on L .

Claim 21 *For all $L \in \mathcal{L}$, the requirements in the definition of continuously limit-effective language standardizable class are satisfied by r , s , and l_L .*

Proof: Claim 19 and 20 respectively imply requirements 1 and 2 in the definition of continuous limit-effective language standardizability. Consider any $L \in \mathcal{L}$. Clearly, D_{r_L} is $\max(D_{r_L})$ -consistent with L (note the definition of D' in step 1 of the definition or r). Consider any j, n such that

- 1) $n \geq l_L$,
- 2) $D_{r(j, n)}$ is $\max(D_{r_L})$ -consistent with L .

We then show that $s(j, n) = s_L$. Clearly, this is true when $L = \emptyset$. Thus, let us assume that $L \neq \emptyset$. Let D, l, D' be as calculated in $r(j, n)$. Now $D_{r_L} \subseteq D_{r(j, n)} \subseteq L$ (by the definition of consistency). Since, $\langle S_L, l_L \rangle$ is a **TextEx** locking sequence for \mathbf{M} on L and $S_L \subseteq D_{r_L} \subseteq D_{r(j, n)} = D' \subseteq L$, we have $\mathbf{M}(\langle D', \text{card}(D') + l + l_L \rangle) = s_L$. In step 2 of the definition of r , it has been checked that $\mathbf{M}(\langle D, l \rangle) = \mathbf{M}(\langle D', \text{card}(D') + l + l_L \rangle)$. Thus, $\mathbf{M}(\langle D, l \rangle) = s_L$ and therefore $s(j, n) = s_L$. ■

From the above claims it follows that \mathcal{L} is continuously limit-effective language standardizable, and hence, **TextEx** \subseteq **Clels**. We now show that

Clels \subseteq **TxtEx**. Let $\mathcal{L} \in \mathbf{Clels}$. We show that $\mathcal{L} \in \mathbf{TxtEx}$. Let r define a limit-effective language operation and s define a standardizing limit-effective language operation as in the definition of continuous limit-effective language standardizability. For each $L \in \mathcal{L}$, let l_L be as defined in the definition of continuous limit-effective language standardizability. We now give the construction of a language learning machine \mathbf{M} which **TxtEx**-identifies \mathcal{L} .

```

begin { $\mathbf{M}(T[n])$ }
  1. Let CandidateSet =  $\{j \mid j \leq n \text{ and } D_{r(j,n)} \text{ is } \max(D_{r(j,n)})\text{-}$ 
    consistent with  $\text{content}(T[n])\}$ ;
  2. if CandidateSet =  $\emptyset$ 
    then output 0
    else output  $s(j, n)$  where  $j = \mu k [k \in \mathbf{CandidateSet} \wedge$ 
     $\max(D_{r(k,n)}) = \max(\{\max(D_{r(i,n)}) \mid i \in \mathbf{CandidateSet}\})]$ 
    endif
end { $\mathbf{M}(T[n])$ }

```

Claim 22 \mathbf{M} **TxtEx**-identifies \mathcal{L} .

Proof: Let $L \in \mathcal{L}$. Let T be a text for L . Let k be such that $W_k = L$. Let n_0 be such that for all $n > n_0$, $r(k, n) = r_L$. Clearly, such an n_0 exists (by definition of continuously limit-effective language standardizability). Let n_1 be so large that the following hold:

- (a) $n_1 \geq l_L$;
- (b) $\text{content}(T[n_1]) \supseteq D_{r_L}$;
- (c) $n_1 > k$; and
- (d) $n_1 > n_0$.

Clearly, such an n_1 exists. Now consider the procedure for $\mathbf{M}(T[n])$, for $n \geq n_1$. k is in the **CandidateSet** (by step 1 in the construction of \mathbf{M}). Let $j \in \mathbf{CandidateSet}$ be such that $\max(D_{r(j,n)}) \geq \max(D_{r_L})$. Since $j \in \mathbf{CandidateSet}$, $D_{r(j,n)}$ is $\max(D_{r(j,n)})$ -consistent with $\text{content}(T[n])$. This implies that $D_{r(j,n)}$ is $\max(D_{r_L})$ -consistent with L . Hence, by the

definition of continuously limit-effective language standardizability, $s(j, n) = s_L$. Therefore, $\mathbf{M}(T[n]) = s_L$. Thus, \mathbf{M} **TxtEx**-identifies \mathcal{L} . ■

This proves Theorem 18. ■

6 Summary

The theory of standardizing operations could be used to gain insights into formal language learning theory. Towards this goal, we have given characterizations of notions about language identification in terms of standardizing operations. We have shown that the natural notion of limit-effective language standardizable operation turns out to be more general than Gold’s seminal notion of **TxtEx**-identification. To characterize **TxtEx**-identification exactly, we have introduced restrictions on the idea of limit-effective language standardizing operation. We also borrow concepts from additional information studies in language learning to characterize limit-effective language standardizing operation in terms of a more general notion than **TxtEx**-identification. Our results can be summarized as follows:

$$\mathbf{TxtEx} = \mathbf{Clels} \subset \mathbf{TxtUniBex} = \mathbf{Lels} \subset \mathbf{TxtBex} \subset 2^{\mathcal{E}}.$$

Freivalds [5], Chen [4], Case, Jain, and Sharma [3] have made use of similar characterizations to gain an insight into the study of program size restrictions in inductive learning. We hope that the results presented here will provide a new way to approach various issues in formal language learning theory.

7 Acknowledgements

We would like to thank John Case, Mark Fulk, and Rajeev Raman for helpful discussions. We are also grateful to an anonymous referee whose comments have resulted in several improvements in the paper. This work was carried out when Sanjay Jain was supported by the NSF grant CCR 832-0136 at

the University of Rochester and Arun Sharma was supported by the NSF grant CCR 871-3846 to John Case at SUNY at Buffalo and the University of Delaware. Finally, we would also like to express our gratitude to Professor S. N. Maheshwari of the Department of Computer Science and Engineering at the Indian Institute of Technology, New Delhi for making the facilities of his department available to us during the preparation of this manuscript.

References

- [1] L. Blum and M. Blum. Toward a mathematical theory of inductive inference. *Information and Control*, 28:125–155, 1975.
- [2] M. Blum. A machine-independent theory of the complexity of recursive functions. *Journal of the ACM*, 14:322–336, 1967.
- [3] J. Case, S. Jain, and A. Sharma. Convergence to nearly minimal size grammars by vacillating learning machines. In R. Rivest, D. Haussler, and M. Warmuth, editors, *Proceedings of the Second Annual Workshop on Computational Learning Theory*, pages 189–199. Morgan Kaufmann, 1989.
- [4] K. J. Chen. Tradeoffs in inductive inference of nearly minimal sized programs. *Information and Control*, 52:68–86, 1982.
- [5] R. Freivalds. Minimal Gödel numbers and their identification in the limit. In *Mathematical Foundations of Computer Science*, volume 32 of *Lecture Notes in Computer Science*, pages 219–225. Springer-Verlag, 1975.
- [6] R. Freivalds, E. Kinber, and R. Wiehagen. Connections between identifying functionals, standardizing operations, and computable numberings. *Zeitschr. j. math. Logik und Grundlagen d. Math. Bd.*, 30:145–164, 1984.

- [7] R. Freivalds, E. Kinber, and R. Wiehagen. Inductive inference from good examples. In *Analogical and Inductive Inference, Proceedings of the Second International Workshop (AII '89)*, volume 397 of *Lecture Notes in Artificial Intelligence*, pages 1–17. Springer-Verlag, 1989.
- [8] M. Fulk. *A Study of Inductive Inference Machines*. PhD thesis, SUNY/Buffalo, 1985.
- [9] M. Fulk. Saving the phenomenon: Requirements that inductive machines not contradict known data. *Information and Computation*, 79:193–209, 1988.
- [10] E. M. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.
- [11] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.
- [12] S. Jain and A. Sharma. Learning in the presence of partial explanations. *Information and Computation*, 95:162–191, 1991.
- [13] S. Jain and A. Sharma. Learning with the knowledge of an upper bound on program size. *Information and Computation*, 102:118–166, 1993.
- [14] E. Kinber. On comparison of limit identification and limit standardization of general recursive functions. *Uch. zap. Latv. univ.*, 233:45–56, 1975.
- [15] D. Osherson, M. Stob, and S. Weinstein. Note on a central lemma of learning theory. *Journal of Mathematical Psychology*, 27:86–92, 1983.
- [16] D. Osherson, M. Stob, and S. Weinstein. *Systems that Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists*. MIT Press, 1986.
- [17] D. Osherson and S. Weinstein. A note on formal learning theory. *Cognition*, 11:77–88, 1982.

- [18] S. Pinker. Formal models of language learning. *Cognition*, 7:217–283, 1979.
- [19] H. Rogers. Gödel numberings of partial recursive functions. *Journal of Symbolic Logic*, 23:331–341, 1958.
- [20] H. Rogers. *Theory of Recursive Functions and Effective Computability*. McGraw-Hill, 1967. Reprinted by MIT Press in 1987.
- [21] G. Schäfer-Richter. *Über Eingabeabhängigkeit und Komplexität von Inferenzstrategien*. PhD thesis, RWTH Aachen, 1984.
- [22] K. Wexler. On extensional learnability. *Cognition*, 11:89–95, 1982.
- [23] K. Wexler and P. Culicover. *Formal Principles of Language Acquisition*. MIT Press, 1980.