Vacillatory Learning of Nearly Minimal Size Grammars

John Case
Department of Computer and Information Sciences
University of Delaware
Newark, DE 19716
USA
Email: case@cis.udel.edu

Sanjay Jain
Institute of Systems Science
National University of Singapore
Singapore 0511
Email: sanjay@iss.nus.sg

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

March 11, 2007

Abstract

In Gold's influential language learning paradigm a learning machine converges in the limit to *one* correct grammar. In an attempt to generalize Gold's paradigm, Case considered the question whether people might converge to vacillating between up to (some integer) $n > 1$ distinct, but equivalent, correct grammars. He showed that larger classes of languages can be algorithmically learned (in the limit) by converging to *up to* $n + 1$ rather than up to n correct grammars. He also argued that, for "small" $n > 1$, it is plausible that people might sometimes converge to vacillating between up to n grammars. The insistence on *small* n was motivated by the consideration that, for "large" n, at least one of n grammars would be too large to fit in peoples' heads. Of course, even for Gold's $n = 1$ case, the single grammar converged to in the limit may be infeasibly large. An interesting complexity restriction to make, then, on the final grammar(s) converged to in the limit is that they all have small size. In this paper we study some of the tradeoffs in learning power involved in making a well-defined version of this restriction.

We show *and exploit as a tool* the desirable property that the learning power under our size-restricted criteria (for successful learning) is independent of the underlying acceptable programming systems. We characterize the power of our size-restricted criteria and use this characterization to prove that *some* classes of languages, which can be learned by converging in the limit to up to $n + 1$ *nearly minimal size* correct grammars, *cannot* be learned by converging to up to n unrestricted grammars even if these latter grammars are allowed to have a finite number of anomalies (i.e., mistakes) per grammar.

We also show that there is *no* loss of learning power in demanding that the final grammars be nearly minimal size *iff* one is willing to tolerate an *unbounded*, finite number of anomalies in the final grammars *and* there is a *constant* bound on the number of different grammars converged to in the limit. Hence, if we allow an unbounded, finite number of anomalies in the final grammars *and* the number of different grammars converged to in the limit is *unbounded* but finite (or if there is a *constant* bound on the number of anomalies allowed in the final grammars), then there *is* a loss of learning power in requiring that the final grammars be nearly minimal size.

These results do not always match what might be expected from the cases, previously examined by Freivalds, Kinber, and Chen, of learning nearly minimal size programs for functions.

1 Preliminaries

Recursion-theoretic concepts not explained below are treated in [30]. N denotes the set of natural numbers, $\{0, 1, 2, 3, \ldots\}$, and I^+ denotes the set of positive integers. Conventions (to follow) as to the range of variables apply to these variables with or without decorations[1]. a and b range over $(N \cup \{*\})$ and $(I^+ \cup \{*\})$, respectively. f, g, h, and v range over (total) functions with arguments and values from N. Other lower case letters near the front and rear of the alphabet range over N. $\subseteq$ denotes the subset relation, and $\subset$ denotes proper subset. $\emptyset$ denotes empty set. $\uparrow$ denotes undefined. $\downarrow$ denotes defined. $\max(S)$ and $\min(S)$ denote maximum and minimum elements of the set S. By convention $\max(\emptyset) = 0$ and $\min(\emptyset) = \uparrow$. $\mathrm{card}(S)$ denotes the cardinality of S. L ranges over subsets of N which are usually construed as codings of formal languages. $L_1 \mathbf{\Delta} L_2$ denotes $(L_1 - L_2) \cup (L_2 - L_1)$, the symmetric difference of L_1 and L_2. We let φ range over acceptable programming systems (numberings) for the partial recursive functions: $N \to N$ [3, 23]. $\mathcal{R}$ denotes the set of all total computable functions. For $i \in N$ and $f \in \mathcal{R}$, we say that $\varphi_i \subseteq f$ just in case $(\forall x)[\varphi_i(x)\downarrow \Rightarrow \varphi_i(x) = f(x)]$. We let Φ denote an arbitrary fixed Blum complexity measure for the system φ. W_p^φ denotes the domain of φ_p. W_p^φ is, then, the r.e. set/language ($\subseteq N$) accepted by φ-program p. We can (and do) also think of p as (coding) a (type 0 [19]) grammar for generating W_p^φ. $W_{p,s}^\varphi = \{x < s \mid \Phi_p(x) < s\}$. We let P range over subsets of N usually construed as sets of φ-programs/grammars. Let $\lambda x, y\,.\langle x, y\rangle$ denote a fixed pairing function (a recursive, bijective mapping: $N \times N \to N$ [30]). $\lambda x, y\,.\langle x, y\rangle$ and its inverses are useful to simulate the effect of having multiple argument functions in the systems φ. $\mathrm{mingrammar}_\varphi(L)$ denotes $\min(\{p \mid W_p^\varphi = L\})$. We adopt the convention that $(\forall i \in N)[i < * < \infty]$; intuitively, $*$ just means unbounded, but finite. $L_1 =^a L_2$ means that $\mathrm{card}(L_1 \mathbf{\Delta} L_2) \leq a$, and $f_1 =^a f_2$ means that $\mathrm{card}(\{x \mid f_1(x) \neq f_2(x)\}) \leq a$. $\mathcal{E}$ denotes the class of all *recursively enumerable* languages $\subseteq N$. We let $\mathcal{L}$ range over subsets of $\mathcal{E}$.

Definition 1 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 sequence, of natural numbers and #'s, is the set of natural numbers in its range, where a text is just the infinite case of such a sequence.

Intuitively, a text for a language is an enumeration or sequential presentation of all the objects in the language with the #'s representing pauses in the listing or presentation of such objects. For example, the only text for the empty language is just an infinite sequence of #'s.

We let T range over texts, and σ and τ range over *finite* sequences (of natural numbers and #'s), i.e., over finite initial segments of texts. $T[s]$ denotes the finite initial segment of T with

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

length s. Hence, domain$(T[s]) = \{z \mid z < s\}$. $|\sigma|$ denotes the number of elements in σ. We say that $\sigma \subseteq \tau$ ($\sigma \subset \tau$) just in case σ is an initial segment (proper initial segment) of τ. Also, we say that $\sigma \subset T$ just in case $\sigma = T[|\sigma|]$. $\sigma \diamond (y)$ denotes the sequence formed by adding y to the end of σ. Thus, if $\sigma' = \sigma \diamond (y)$, then, for all x,

$$\sigma'(x) = \begin{cases} \sigma(x), & \text{if } x < |\sigma|; \\ y, & \text{if } x = |\sigma|; \\ \uparrow, & \text{otherwise.} \end{cases}$$

Definition 2 A *learning function* is a computable mapping from the set of all finite sequences, of natural numbers and #'s, into N.

We let $\mathbf{F}$ range over learning functions, and we think of $\mathbf{F}(\sigma)$ as the (Gödel number of) a grammar (based on the fixed acceptable programming system φ). We take $\mathbf{F}(\sigma)$ to be $\mathbf{F}$'s conjecture based on the finitely much data in σ. We let $\mathbf{F}[\tau, \sigma]$ denote the set $\{\mathbf{F}(\tau') \mid \tau \subseteq \tau' \subseteq \sigma\}$. Suppose T is any text for a language L. We are interested in the extent to which, *for sufficiently large* s, the grammars $\mathbf{F}(T[s])$ generate L.

We consider more specifically what it means for a learning function to be successful on a language. Gold [18] essentially proposed the following criterion of success (Definition 3) which we call **TxtEx**-*identification* after [11] (the nomenclature in which was based on that in [12]). The quantifiers '$\forall^\infty$' and '$\exists^\infty$' mean 'for all but finitely many' and 'there exist infinitely many', respectively. The concepts introduced in Definitions 3, 6, and 7 below are *implicitly* parameterized by the choice of acceptable programming system φ and the corresponding programs and grammars.

Definition 3 **F TxtEx**-*identifies* $L \iff (\forall$ texts T for $L)(\exists p \mid W_p^\varphi = L)(\forall^\infty s)[\mathbf{F}(T[s]) = p]$.

Essentially the concepts from Definitions 1, 2, and 3 constitute Gold's influential language learning paradigm discussed, for example, in [26, 29, 33, 32]. In an attempt to generalize Gold's paradigm, Case in [5] considered the question whether people converge to vacillating between up to (some integer) $n > 1$ distinct, but equivalent, correct grammars. It was shown there that larger classes of languages can be algorithmically learned (in the limit) by converging to *up to* $n + 1$ rather than up to n equivalent, correct grammars. He argued that, for "small" $n > 1$, it is plausible that people might sometimes converge to vacillating between up to n grammars. Gold's paradigm allows for convergence to only *one* grammar in the limit.

In the next section, we define appropriate notions from [5] and state some important results.

2 Language Learning by Vacillating Machines

In Definition 4 just below we spell out what it means for a learning function on a text to converge in the limit to a finite set of grammars.

Definition 4 Suppose $\mathbf{F}$ is a learning function and T is a text. $\mathbf{F}(T)\Downarrow$ (read: T *stabilizes* $\mathbf{F}$) $\iff$ $\{\mathbf{F}(\tau) \mid \tau \subset T\}$ is finite. If $\mathbf{F}(T)\Downarrow$, then $\mathbf{F}(T)$ is defined $= \{p \mid (\exists^{\infty}\tau \subset T)[\mathbf{F}(\tau) = p]\}$; otherwise, we say that $\mathbf{F}(T)\Uparrow$.

The following definition is a refinement of Definition 4.

Definition 5 Suppose $\mathbf{F}$ is a learning function and T is a text. Suppose $b \in N^{+} \cup \{*\}$. $\mathbf{F}(T)\Downarrow_b$ (read: T b-*stabilizes* $\mathbf{F}$) $\iff$ $F(T)\Downarrow \wedge \mathrm{card}(\mathbf{F}(T)) \leq b$. If $\neg\mathbf{F}(T)\Downarrow_b$, then we say that $\mathbf{F}(T)\Uparrow_b$.

Clearly, $\mathbf{F}(T)\Downarrow$ and $\mathbf{F}(T)\Downarrow_*$ are the same notions. The next definition describes the criteria of vacillatory identification of languages.

Definition 6 Let $b \in N^{+} \cup \{*\}$. A language learning function, $\mathbf{F}$, is said to $\mathbf{TxtFex}_b^a$-*identify* a language L $\iff$ $(\forall$ texts T for $L)[\mathbf{F}(T)\Downarrow_b \wedge (\forall p \in \mathbf{F}(T))[W_p^{\varphi} =^a L]]$.

In $\mathbf{TxtFex}_b^a$-identification the b is a "bound" on the number of final grammars and the a is a "bound" on the number of anomalies allowed in any of these final grammars. A "bound" of $*$ just means unbounded, but finite.

Definition 7 $\mathbf{TxtFex}_b^a$ denotes the class of all sets $\mathcal{L}$ of languages such that some learning function $\mathbf{TxtFex}_b^a$-identifies each language in $\mathcal{L}$.

$\mathbf{TxtFex}_b^a$ provides a set-theoretic summary of the power of individual learning functions to $\mathbf{TxtFex}_b^a$-identify entire classes of languages. It is easy to show that it does *not* depend on the particular choice of acceptable programming system φ on which it is based.

We proceed (Definition 8) to describe an interesting and useful restriction on learning functions which generalizes notions of *order independence* from [2, 26, 16, 17].

Definition 8 [5] A learning function, $\mathbf{F}$, is *b-ary order independent* $\iff$ $(\forall L \mid$ some text for L b-stabilizes $\mathbf{F})[(\exists P$ of cardinality $\leq b)(\forall$ texts T for $L)[\mathbf{F}(T)\Downarrow_b = P]]$.

The following result from [6] is a generalization of results from [2, 26, 16, 17]. It is also an ostensibly indispensable tool for proving Theorems 17 and 25.

Theorem 9 [6] *There is an algorithm for transforming any b and any program for a learning function $\mathbf{F}$ into a corresponding program for a learning function $\mathbf{F}'$ such that $\mathbf{F}'$ is b-ary order independent and $(\forall a)(\forall L)[\mathbf{F}$ $\mathbf{TxtFex}_b^a$-identifies $L \Rightarrow \mathbf{F}'$ $\mathbf{TxtFex}_b^a$-identifies $L]$.*

3 Convergence to Nearly Minimal Size Grammars

As noted above Case [5] argued that, for "small" n, it is plausible that people might sometimes converge to vacillating between up to n grammars. The insistence on *small* n was motivated by the consideration that, for "large" n, at least one of n grammars would be too large to fit in our heads. This latter assumes, of course, that human brain storage is not magic, admitting of infinite regress, etc. Of course, even for Gold's $n = 1$ case, the single grammar converged to in the limit may be infeasibly large. An interesting complexity restriction to make, then, on the final grammar(s) converged to in the limit is that they all have small size. In this paper we study some of the tradeoffs in learning power involved in making such a reasonable restriction on the $\mathbf{TxtFex}_b^a$ criteria.

Freivalds [15], and later Kinber [22] and Chen [13, 14], considered the case of learning small size programs for computable functions. Case and Chi [8] consider the case of inferring small size grammars within the context of Gold's paradigm. Jain and Sharma [20, 21] show that within this latter context the severe restriction of requiring the final grammar to be absolutely minimal size produces a learning criterion *dependent* on the choice of acceptable programming system φ. Freivalds previously [15] obtained a similar result in the context of learning minimal size programs for computable functions. Generally, *strictly* minimal size programs or grammars are hard to deal with[2], and information-theoretic considerations suggest that such objects may be so deficient in information content [31] as to be difficult to understand (even subconsciously). Freivalds [15] invented a mathematically elegant, precise notion of *nearly* minimal size programs—again in the context of learning programs for functions. In this paper we study the extension to vacillatory learning, i.e., to the $\mathbf{TxtFex}_b^a$ learning criteria.[3] This extension turns out to be non-trivial, its study ostensibly requiring the invention of new tools. Furthermore, the results do not always match what might be expected from the case of learning programs for functions. For convenience the concepts introduced in Definitions 10 and 12 below are *explicitly* parameterized by the choice of acceptable system φ.

Definition 10 A language learning function, $\mathbf{F}$, $\mathbf{TxtMfex}_b^a$-*identifies* a class of languages $\mathcal{L}$ *in the* φ*-programming system* $\iff$ $(\exists$ recursive $h)(\forall L \in \mathcal{L})[\mathbf{F}$ $\mathbf{TxtFex}_b^a$-identifies L in the φ-programming system $\wedge$ $(\forall$ texts T for $L)(\forall p \in \mathbf{F}(T))[p \leq h(\text{mingrammar}_\varphi(L))]]$.

h in Definition 10 plays the role of a *computable* amount by which the final programs can

[2] For example, $\{i \mid i = \text{mingrammar}_\varphi(W_i)\}$ is an immune set (see [30]).

[3] Essentially, this extension for the $b = *$ case turns out to be equivalent to applying program size restriction ideas to behaviorally correct language identification; these issues are discussed in Section 6.

be larger than minimal size. Even this size restriction of course does not hold in general, but it is not as severe as requiring that the final programs be strictly minimal size.

Remark 11 *It is easy to argue that we may always take h to be monotone increasing in Definition 10.*

Mathematically, $\mathbf{TxtMfex}_b^a$-identification is well-behaved. For example, Proposition 13 below asserts that it is *independent* of the choice of the acceptable programming system φ. First we provide in Definition 12 a notation (analogous to that of Definition 7) providing a set-theoretic summary of the power of individual learning functions to $\mathbf{TxtMfex}_b^a$-identify entire classes of languages *in the φ programming system.*

Definition 12 $\mathbf{TxtMfex}_b^a(\varphi) =$

$$\{\mathcal{L} \mid (\exists \mathbf{F})[\mathbf{F}\ \mathbf{TxtMfex}_b^a\text{-identifies } \mathcal{L} \text{ in the } \varphi\text{-programming system}]\}.$$

Proposition 13 just below says that the power of $\mathbf{TxtMfex}_b^a$-identification is independent of the choice of acceptable programming system. This is a desirable result in its own right *and* is a useful tool to prove other results. There is an analogous result regarding the learning of nearly minimal size programs for functions in [15, 13, 14].

Proposition 13 $(\forall \varphi, \varphi')[\mathbf{TxtMfex}_b^a(\varphi) = \mathbf{TxtMfex}_b^a(\varphi')]$.

Proof. Suppose $\mathbf{F}$ $\mathbf{TxtMfex}_b^a$-identifies $\mathcal{L}$ in the φ programming system. Let h be such that for all texts T for $L \in \mathcal{L}$, $\mathbf{F}(T) \leq h(\text{mingrammar}_\varphi(L))$. By Remark 11 we may assume, without loss of generality, that h is monotone increasing.

Let g and g' be monotone increasing, recursive functions such that for all i, $\varphi_{g(i)} = \varphi'_i$ and $\varphi'_{g'(i)} = \varphi_i$. Note that, for each L, $\text{mingrammar}_\varphi(L) \leq g(\text{mingrammar}_{\varphi'}(L))$.

Define $\mathbf{F}'$ as follows. $\mathbf{F}'(\sigma) = g'(\mathbf{F}(\sigma))$. Clearly, $\mathbf{F}'$ $\mathbf{TxtFex}_b^a$-identifies $\mathcal{L}$. Now consider a text T for $L \in \mathcal{L}$. Let $j \in \mathbf{F}'(T)$. Clearly, there exists an $j_{\mathbf{F}} \in \mathbf{F}(T)$, such that $j = g'(j_{\mathbf{F}})$. Also $j = g'(j_{\mathbf{F}}) \leq g'(h(\text{mingrammar}_\varphi(L))) \leq g'(h(g(\text{mingrammar}_{\varphi'}(L))))$. It follows that $\mathbf{F}'$ $\mathbf{TxtMfex}_b^a$-identifies $\mathcal{L}$. □

From now on it is permissible to write $\mathbf{TxtMfex}_b^a$ for $\mathbf{TxtMfex}_b^a(\varphi)$, and we do so.

4 A Characterization of $\mathbf{TxtMfex}_b^a$ Criteria

We introduce below (Definition 15) an intrinsically interesting and technically useful notion that will help us formulate a characterization of $\mathbf{TxtMfex}_b^a$. To that end, it is useful to first

introduce Definition 14 which provides an interesting new extension of the ordinary notion of limit.

Definition 14 Suppose g is a recursive function in two variables and $b \in N \cup \{*\}$. Then we say $\lim_s g(i,s)\Downarrow_b \iff (\exists P \text{ of cardinality } \leq b)[(\forall^\infty s)[g(i,s) \in P] \wedge (\forall p \in P)(\exists^\infty s)[g(i,s) = p]]$. If $\lim_s g(i,s)\Downarrow_b$, we define $b\text{-}\lim_s g(i,s) = \{p \mid (\exists^\infty s)[g(i,s) = p]\}$; otherwise, $b\text{-}\lim_s g(i,s)$ is undefined.

Definition 15 provides a nice extension of a concept originating with Freivalds [15] (and later studied and extended by Chen [13, 14]) in the context of the learning of nearly minimal size programs for functions.

Definition 15 $\mathcal{L}$ is *text (a,b)-standardizable with a recursive estimate* (abbreviated: $\mathcal{L} \in \mathbf{TxtFlsr}^a_b$) $\iff$ there exist recursive functions g and v such that, for all $L \in \mathcal{L}$ and $i \in N$, if $W^\varphi_i = L$, then

(a) $\lim_s g(i,s)\Downarrow_b$ and $(\forall p \in b\text{-}\lim_s g(i,s))[W^\varphi_p =^a L]$;

(b) for all j, if $W^\varphi_j = L$, then $\lim_s g(j,s)\Downarrow_b$ and $b\text{-}\lim_s g(i,s) = b\text{-}\lim_s g(j,s)$;

(c) $\text{card}(\{g(i,s) \mid s \in N\}) \leq v(i)$.

If recursive functions g and v witness that $\mathcal{L} \in \mathbf{TxtFlsr}^a_b$, then we write $\mathcal{L} \subseteq \mathbf{TxtFlsr}^a_b(g,v)$. It is easy to verify that $\mathbf{TxtFlsr}^a_b$ is acceptable programming system independent.

We give some intuitive insight into $\mathbf{TxtFlsr}^a_b$. It will suffice to consider the $a = 0$ and $b = 1$ cases. The *grammar equivalence problem* $(\{\langle x,y\rangle \mid W^\varphi_x = W^\varphi_y\})$ is well-known to be Π^0_2-complete [30]; hence, it cannot be *accepted* by a *limiting* recursive procedure. The role of g in the definition of $\mathbf{TxtFlsr}^0_1$ is to indirectly provide a limiting recursive solution to this problem for the special cases where the grammars generate languages in $\mathcal{L}$: g finds (in the limit) *canonical* grammars. Also, v places some extra constraint on how g reaches its limits.

Convention 1 *For the rest of the paper we take φ to be a fixed acceptable programming system. From now on we usually write* 'mingrammar' *for* 'mingrammar$_\varphi$' *and W_i for W^φ_i.*

Theorem 17 below is a characterization of $\mathbf{TxtMfex}^a_b$. Our proof of this theorem ostensibly requires, in addition to Theorem 9, the following variant of Theorem 9.

Theorem 16 [6] *There is an algorithm for transforming any b and any program for a learning function $\mathbf{F}$ into a corresponding program for a learning function $\mathbf{F}'$ such that $\mathbf{F}'$ is b-ary order independent and $(\forall a)(\forall \mathcal{L})[\mathbf{F}$ $\mathbf{TxtMfex}^a_b$-identifies $\mathcal{L} \Rightarrow \mathbf{F}'$ $\mathbf{TxtMfex}^a_b$-identifies $\mathcal{L}]$.*

The following theorem has analogs [15, 13, 14] for the learning of programs for functions.

Theorem 17 *The following three statements are equivalent.*

(1) $\mathcal{L} \in \mathbf{TxtMfex}^a_b$.

(2) There exist recursive functions, g *and* v*, a language learning function* $\mathbf{F}$ *such that* $\mathcal{L} \subseteq \mathbf{TxtFlsr}^a_b(g, v)$*,* $\mathbf{F}$ $\mathbf{TxtFex}^a_b$*-identifies* $\mathcal{L}$ *and* $(\forall i \mid W_i \in \mathcal{L})(\forall$ *texts* T *for* $W_i)[\mathbf{F}(T) = b\text{-}\lim_s g(i, s)]$.

(3) There exist recursive functions, g *and* v*, and a language learning function* $\mathbf{F}$ *such that* $\mathbf{F}$ $\mathbf{TxtFex}^a_b$*-identifies* $\mathcal{L}$ *and* $(\forall L \in \mathcal{L})[(\forall$ *texts* T *for* $L)[\mathbf{F}(T) = b\text{-}\lim_s g(\text{mingrammar}(L), s)] \wedge [\text{card}(\{g(\text{mingrammar}(L), s) \mid s \in N\}) \leq v(\text{mingrammar}(L))]]$.

Proof of Theorem 17.

(1) $\Rightarrow$ (2). Suppose $\mathbf{F}$ $\mathbf{TxtMfex}^a_b$-identifies $\mathcal{L}$. Let h be such that, for all $L \in \mathcal{L}$, for all texts T for L, for each $i \in \mathbf{F}(T)$, $i \leq h(\text{mingrammar}(L))$. By Theorem 16, without loss of generality, we may take $\mathbf{F}$ to be b-ary order independent. Also, by Remark 11, without loss of generality we may take h to be monotone increasing. Let T_i be a text for W_i, such that $T_i[s]$ can be effectively computed from s and i. Let $v(i) = h(i) + 1$ and define g as follows:

$$g(i, s) = \begin{cases} \mathbf{F}(T_i[s]), & \text{if } \mathbf{F}(T_i[s]) \leq h(i); \\ 0, & \text{otherwise.} \end{cases}$$

It is clear that both g and v are recursive. We first show that $(\forall i \mid W_i \in \mathcal{L})(\forall$ texts T for $W_i)[\mathbf{F}(T) = b\text{-}\lim_s g(i, s)]$. Let $W_i \in \mathcal{L}$. Then, according to the definition of g, $b\text{-}\lim_s g(i, s) = \mathbf{F}(T_i)$. Furthermore, since $\mathbf{F}$ is a b-ary order independent language learning function, we have $b\text{-}\lim_s g(i, s) = \mathbf{F}(T)$, for any text T for W_i.

We now show that $\mathcal{L} \subseteq \mathbf{TxtFlsr}^a_b(g, v)$. Suppose $L \in \mathcal{L}$ and $W_i = L$. Then T_i is a text for L, and, since $\mathbf{F}$ $\mathbf{TxtMfex}^a_b$-identifies $\mathcal{L}$, we have $(\forall p \in \mathbf{F}(T_i))[W_p =^a L \wedge p \leq h(\text{mingrammar}(L)) \leq h(i)]$. Thus, by the definition of g, $\lim_s g(i, s)\Downarrow_b$ and $b\text{-}\lim_s g(i, s) = \mathbf{F}(T_i)$ and $(\forall p \in b\text{-}\lim_s g(i, s))[W_p =^a L]$. Furthermore, if $W_j = L$, then $b\text{-}\lim_s g(j, s) = \mathbf{F}(T_j) = b\text{-}\lim_s g(i, s)$, since T_j is also a text for L and $\mathbf{F}$ is b-ary order independent. Also, for any $W_i \in \mathcal{L}$, $\{g(i, s) \mid s \in N\} \subseteq \{k \mid k \leq h(i)\} \cup \{0\}$. Hence, $\text{card}(\{g(i, s) \mid s \in N\}) \leq h(i) + 1 = v(i)$. Therefore, $\mathcal{L} \subseteq \mathbf{TxtFlsr}^a_b(g, v)$.

(2) $\Rightarrow$ (3). Immediate, since $\text{mingrammar}(L)$ is one of the grammars for L.

(3) $\Rightarrow$ (1). Suppose that, g, v and $\mathbf{F}$ are such that g, v are recursive, and, for every $L \in \mathcal{L}$, (i) $L \in \mathbf{TxtFex}^a_b(\mathbf{F})$, (ii) $(\forall$ texts T for $L)[b\text{-}\lim_s g(\text{mingrammar}(L), s) = \mathbf{F}(T)]$ and (iii) $\text{card}(\{g(\text{mingrammar}(L), s) \mid s \in N\}) \leq v(\text{mingrammar}(L))$. Without loss of generality, we assume that for every i, $\text{card}(\{g(i, s) \mid s \in N\}) \leq v(i)$.

By the *s-m-n theorem* [30], there is a recursive function z such that for all i, j,

$$W_{z(i,j)} = \begin{cases} W_p, & \text{if there exists an } m \text{ such that} \\ & \text{card}(\{g(i,k) \mid k \leq m\}) = j \\ & \text{and } p = g(i,m) \text{ for the least such } m; \\ \emptyset, & \text{otherwise.} \end{cases}$$

We define $\mathbf{F}'$ thus. Suppose T is an arbitrary text.

$$\mathbf{F}'(T[x]) = \begin{cases} z(i_l, j), & \text{if } (\exists i \leq x)(\exists y \leq x)[g(i,y) = \mathbf{F}(T[x])] \text{ and} \\ & i_l = \min(\{i \leq x \mid (\exists y \leq x)[g(i,y) = \mathbf{F}(T[x])]\}) \text{ and} \\ & j = \text{card}(\{g(i_l,k) \mid (\forall k' < k)[g(i_l,k') \neq \mathbf{F}(T[x])]\}); \\ 0, & \text{otherwise.} \end{cases}$$

Let

$$h(i) = \max(\{z(k,l) \mid k \leq i \text{ and } l \leq v(k)\}).$$

Suppose T is a text for $L \in \mathcal{L}$. Let $\text{card}(\mathbf{F}(T)) = b' \leq b$. Let $p_1, p_2, \cdots, p_{b'}$ be members of $\mathbf{F}(T)$. Let i_n, $1 \leq n \leq b'$, be the b' least integers for which there exist a y such that $g(i_n, y) = p_n$. Let y_n, $1 \leq n \leq b'$, be the least y such that $g(i_n, y_n) = p_n$. Let $I = \{i_n \mid 1 \leq n \leq b'\}$ and $Y = \{y_n \mid 1 \leq n \leq b'\}$.

For any $L \in \mathcal{L}$ and any text T for L, let x_0 be a sufficiently large number such that

(i) $(\forall x \geq x_0)[\mathbf{F}(T[x]) \in \mathbf{F}(T)]$ and

(ii) $x_0 \geq \max(\{\max(I), \max(Y)\})$.

Since, $b\text{-lim}_s\, g(\text{mingrammar}(L), s) = \mathbf{F}(T)$, we observe that $(\forall i \in I)[i \leq \text{mingrammar}(L)]$. Also, by the definition of $\mathbf{F}'$, for all $x \geq x_0$, $\mathbf{F}'(T[x]) = z(i_n, j_n)$, where $j_n = \text{card}(\{g(i_n, y) \mid y \leq y_n\})$ and $i_n \in I$ and $y_n \in Y$ correspond to the program $\mathbf{F}(T[x]) \in \mathbf{F}(T)$. Furthermore, $(\forall n \mid 1 \leq n \leq b')[W_{z(i_n,j_n)} = W_{g(i_n,y_n)} \wedge [(\exists p \in \mathbf{F}(T))[W_{z(i_n,j_n)} = W_p]]]$. Hence, $(\forall n \mid 1 \leq n \leq b')[W_{z(i_n,j_n)} =^a L]$. Since, $(\forall n \mid 1 \leq n \leq b')[i_n \leq \text{mingrammar}(L)]$ and $(\forall n \mid 1 \leq n \leq b')[j_n \leq v(i_n)]$, we have $(\forall n \mid 1 \leq n \leq b')[z(i_n, j_n) \leq h(\text{mingrammar}(L))]$. Thus, $\mathcal{L} \in \mathbf{TxtMfex}^a_b$. ∎

The following useful corollaries to Theorem 17 involve variants of a self-referential class from [5]. We could not make it go for the self-referential class from [5]. Chen [13, 14] made direct use of simpler self-referential classes from [12] to obtain a useful analog of Corollaries 18 and 19—but for the problem of learning programs for functions.

Corollary 18 *Suppose $n > 0$. Let $\mathcal{L}_n = \{L \mid L \text{ is } \infty \wedge (\forall^\infty \langle x, y\rangle \in L)[W_y = L] \wedge [\text{card}(\{y \mid (\exists x)[\langle x, y\rangle \in L]\}) \leq n]\}$. Then $\mathcal{L}_n \in \mathbf{TxtMfex}^0_n$.*

Corollary 19 *Let $\mathcal{L}_* = \{L \mid L \text{ is } \infty \wedge (\forall^\infty \langle x, y\rangle \in L)[W_y = L] \wedge [\text{card}(\{y \mid (\exists x)[\langle x, y\rangle \in L]\}) \leq \text{mingrammar}(L)]\}$. Then $\mathcal{L}_* \in \mathbf{TxtMfex}^0_*$.*

5 Comparison of Learning with and without Size Restrictions

Using Corollary 18 and a modification of a multiple recursion theorem argument from [5], we show the following result which implies that some classes of languages can be algorithmically learned (in the limit) by converging to *up to* $n+1$ *nearly minimal size* grammars but *cannot* be learned by converging to up to n *unrestricted* grammars even if these latter grammars are allowed to have a finite number of anomalies per grammar.

Theorem 20 *Suppose* $n > 0$. *Then* $\mathbf{TxtMfex}^0_{n+1} - \mathbf{TxtFex}^*_n \neq \emptyset$.

Corollary 21 $\mathbf{TxtMfex}^a_1 \subset \cdots \subset \mathbf{TxtMfex}^a_n \subset \mathbf{TxtMfex}^a_{n+1} \subset \cdots \subset \mathbf{TxtMfex}^a_*$.

Proof of Theorem 20. For $n \in N$, define $\mathcal{L}_{n+1} = \{L \in \mathcal{E} \mid [L \text{ is } \infty] \wedge [\text{card}(\{y \mid (\exists x)[\langle x, y\rangle \in L]\}) \leq n+1] \wedge [(\forall^\infty \langle x, y\rangle \in L)[W_y = L]]\}$.

Clearly, by Corollary 18, $\mathcal{L}_{n+1} \in \mathbf{TxtMfex}^0_{n+1}$. It remains to show that $\mathcal{L}_{n+1} \notin \mathbf{TxtFex}^*_n$.

Suppose by way of contradiction that $\mathbf{F}$ $\mathbf{TxtFex}^*_n$-identifies $\mathcal{L}_{n+1}$. Then, by a padded version of the $(n+1)$-ary recursion theorem there are *distinct* self-other referential $e_1, e_2, \cdots, e_{n+1}$ defining $W_{e_1}, W_{e_2}, \ldots, W_{e_{n+1}}$, respectively as follows.

Informally we let $\mathbf{Last}_n(\mathbf{F}, \sigma)$ denote the set of the last n distinct grammars output by $\mathbf{F}$ when it is fed σ (if the number of distinct grammars output by $\mathbf{F}$ on σ is less than n, then we let $\mathbf{Last}_n(\mathbf{F}, \sigma)$ be the set of grammars output by $\mathbf{F}$ on σ).

Formally, define $\mathbf{Last}_n(\mathbf{F}, T[x]) = \mathbf{F}[T[m], T[x]]$, where $m = \min(\{x' \leq x \mid \text{card}(\mathbf{F}[T[x'], T[x]]) \leq n\})$.

Let $\sigma_0 = \emptyset$. Go to stage 0.

Begin stage s

for $i := 1$ **to** $n+1$ **do**

enumerate $\langle s, e_1\rangle$ in W_{e_i}.

endfor;

Let $\sigma = \sigma_s \diamond (\langle s, e_1\rangle)$.

Dovetail steps 1 and 2 below until, if ever, step 1 succeeds. If and when step 1 succeeds, go to step 3.

1. Search for a τ extending σ such that

   content$(\tau) \subseteq \{\langle x, y\rangle \mid y \in \{e_1, e_2, \cdots, e_{n+1}\}\}$ and
   $\mathbf{Last}_n(\mathbf{F}, \tau) \neq \mathbf{Last}_n(\mathbf{F}, \sigma)$.

2. Go to substage 0.

   Begin substage s'

Enumerate $\langle s', e_j \rangle$ into W_{e_j}, $j \in \{1, 2, \cdots, n+1\}$.

Go to substage $s' + 1$.

End substage s'

3. *(∗ step 1 succeeds. ∗)*

Let $S = \text{content}(\tau) \cup \bigcup_{j \in \{1,2,\cdots,n+1\}} [W_{e_j}$ enumerated till now$]$.

Enumerate S in W_{e_j}, $j \in \{1, 2, \cdots, n+1\}$.

Let σ_{s+1} be an extension of τ such that $\text{content}(\sigma_{s+1}) = S$.

Go to stage $s + 1$.

End stage s

We have the following two cases.

Case 1: Each stage terminates.

Then, $W_{e_1} = W_{e_2} = W_{e_3} = \cdots = W_{e_{n+1}}$. Let $L = W_{e_1}$. Let $T = \cup_{s \in N} \sigma_s$. Clearly T is a text for L and $L \in \mathcal{L}_{n+1}$. But, $\mathbf{F}(T)\Uparrow_n$. Hence, $\mathbf{F}$ does not $\mathbf{TxtFex}^*_n$-identify $\mathcal{L}_{n+1}$.

Case 2: Some stage s starts but does not terminate.

Let $L_j = W_{e_j}$, $j \in \{1, 2, \cdots, n+1\}$. Clearly, for each $j \in \{1, 2, \cdots, n+1\}$, $W_{e_j} \in \mathcal{L}_{n+1}$. Also, for each $j \in \{1, 2, \cdots, n+1\}$, $\text{content}(\sigma_s \diamond (\langle s, e_1 \rangle)) \subseteq L_j$. Since step 1 does not succeed in stage s, for all extensions τ of $\sigma_s \diamond (\langle s, e_1 \rangle)$ such that $\text{content}(\tau) \subseteq \{\langle x, y \rangle \mid y \in \{e_1, \cdots, e_{n+1}\}\}$, $\mathbf{F}(\tau) \in \mathbf{Last}_n(\mathbf{F}, \sigma_s \diamond (\langle s, e_1 \rangle))$. Moreover, for $1 \leq j_1 < j_2 \leq n+1$, $L_{j_1} \neq^* L_{j_2}$. Thus, there exists $j \in \{1, 2, \cdots, n+1\}$ such that, for all $i \in \mathbf{Last}_n(\mathbf{F}, \sigma_s)$, $L_j \neq^* W_i$. $\mathbf{F}$, thus does not $\mathbf{TxtFex}^*_n$-identify $L_j \in \mathcal{L}_{n+1}$.

The above two cases imply the theorem. ∎

We can show that, in the context of the learning of nearly minimal size programs for (total) *functions*, the analog of the hierarchy of Corollary 21 collapses [10]. This result complements results in [1, 12] and answers an open problem in [13].

Theorem 22 $\mathbf{TxtMfex}^0_* - \bigcup_n \mathbf{TxtFex}^*_n \neq \emptyset$.

Proof of Theorem 22. Let $\mathcal{L}_* = \{L \mid L \text{ is } \infty \wedge (\forall^\infty \langle x, y \rangle \in L)[W_y = L] \wedge [\text{card}(\{y \mid (\exists x)[\langle x, y \rangle \in L]\}) \leq \text{mingrammar}(L)]\}$. By Corollary 19, $\mathcal{L}_* \in \mathbf{TxtMfex}^0_*$. Suppose by way of contradiction that $\mathbf{F}$ $\mathbf{TxtFex}^*_n$-identifies $\mathcal{L}_*$. Let $S' = \{j \leq n \mid \text{card}(\{y \mid (\exists x)[\langle x, y \rangle \in W_j]\}) \leq n+1\}$. Let $m' = \max(\{y \mid (\exists j \in S')(\exists x)[\langle x, y \rangle \in W_j]\})$. The rest of the diagonalization is the same as the diagonalization in the proof of Theorem 20, except for the fact that the $n+1$

distinct grammars, $e_1, e_2, \ldots, e_{n+1}$, obtained using the $n+1$-ary recursion theorem must be such that e_1 is padded $> m'$. We leave the details to the reader. ∎

We exploit our characterization theorem (Theorem 17 above) and adapt anomaly hierarchy results from [12] to obtain Theorem 23 below.

Theorem 23 $\mathbf{TxtMfex}_1^{m+1} - \mathbf{TxtFex}_*^m \neq \emptyset$.

Corollary 24 $\mathbf{TxtMfex}_b^1 \subset \mathbf{TxtMfex}_b^2 \subset \cdots \subset \mathbf{TxtMfex}_b^*$.

Proof of Theorem 23. For a recursive function f, define $L_f = \{\langle x, y\rangle \mid f(x) = y\}$. Let $\mathcal{C} = \{f \mid \varphi_{f(0)} =^{m+1} f \wedge \varphi_{f(0)} \subseteq f\}$. Let $\mathcal{L} = \{L_f \mid f \in \mathcal{C}\}$. We claim that $\mathcal{L} \in \mathbf{TxtMfex}_1^{m+1} - \mathbf{TxtFex}_*^m$. Let w be a recursive function such that $W_{w(i)} = \{\langle x, y\rangle \mid \varphi_i(x) = y\}$.

Define $\mathbf{F}, g$ and v as follows.

$$\mathbf{F}(\sigma) = \begin{cases} w(\min(\{y \mid \langle 0, y\rangle \in \text{content}(\sigma)\})), & \text{if } (\exists y)[\langle 0, y\rangle \in \text{content}(\sigma)]; \\ 0, & \text{otherwise.} \end{cases}$$

$$g(i, s) = \begin{cases} w(\min(\{y \mid \langle 0, y\rangle \in \text{content}(W_{i,s})\})), & \text{if } (\exists y)[\langle 0, y\rangle \in W_{i,s}]; \\ 0, & \text{otherwise.} \end{cases}$$

$$v(i) = 2.$$

It is easy to see that

$\mathbf{F}$ $\mathbf{TxtFex}_1^{m+1}$-identifies $\mathcal{L}$,
g, v witness that $\mathcal{L} \in \mathbf{TxtFlsr}_1^{m+1}$ and
$(\forall i \mid W_i \in \mathcal{L})(\forall \text{ texts } T \text{ for } W_i)[\mathbf{F}(T)\Downarrow_1 \ \wedge \ \lim_s g(i,s)\Downarrow_1 \ \wedge \ \mathbf{F}(T) = 1\text{-}\lim_s g(i,s)]$.

Thus, using Theorem 17, we have that $\mathcal{L} \in \mathbf{TxtMfex}_1^{m+1}$. The proof of $\mathcal{C} \notin \mathbf{Ex}^m$ (in [12]) can be easily modified to show that $\mathcal{L} \notin \mathbf{TxtFex}_*^m$. ∎

Theorem 25, to follow, says that, *if* we are willing to tolerate an *unbounded* finite number of anomalies in final grammars, then, as long as there is a *constant* bound on the number of different grammars converged to in the limit, there is *no* loss of learning power in demanding that the final grammars be nearly minimal size. This theorem is possibly the technically hardest to prove in the paper. Although Chen [13, 14] has an analogous result regarding learning programs for functions, his proof depends on exploiting the totality and single-valuedness of recursive functions. We apply Theorem 9 and Proposition 13 and employ a new combinatorial trick.

Theorem 25 *Suppose* $n > 0$. *Then* $\mathbf{TxtMfex}_n^* = \mathbf{TxtFex}_n^*$.

PROOF OF THEOREM 25. We need to show that, for any n, $\mathbf{TxtFex}^*_n \subseteq \mathbf{TxtMfex}^*_n$. Let $\mathbf{F}$ be any learning function which $\mathbf{TxtFex}^*_n(\mathbf{F})$-identifies $\mathcal{L}$. Then, by Proposition 13, it suffices to define an acceptable programming system φ' and a learning function $\mathbf{F}'$, such that $\mathcal{L}$ is $\mathbf{TxtMfex}^*_n$-identified by $\mathbf{F}'$ in the φ'-programming system.

By Theorem 9, without loss of generality, we assume that $\mathbf{F}$ is n-ary order independent. Let T_j denote a text for the r.e. language W_j such that $T_j[x]$, the finite initial segment of T_j of length x, can be recursively obtained from j and x.

Let $\mathbf{Last}_n(\mathbf{F}, \sigma)$ be as defined in the proof of Theorem 20. Let $S^x_j = \mathbf{Last}_n(\mathbf{F}, T_j[x])$. Let $G^{x,1}_j < G^{x,2}_j < \cdots < G^{x,\mathrm{card}(S^x_j)}_j$ be the elements of S^x_j. Let $\mathbf{PreviousMindChange}(j,x) = \max(\{k' \leq x \mid \mathbf{F}(T_j[k']) \notin S^x_j\})$. We now define, effectively in k, $W^{\varphi'}_k$ (via an enumeration procedure).

Begin { procedure for enumerating $W^{\varphi'}_k$ }

if $(\exists j)[k = (n+1)j]$

then *(∗ this helps to make φ' acceptable ∗)*

Let $W^{\varphi'}_k = W^{\varphi}_j$.

else

Let p and r be such that $r \leq n$ and $k = (n+1)p + r$.

Go to stage 0.

Begin stage s

For $q = G^{s,r}_p$, put $W^{\varphi}_{q,s}$ into $W^{\varphi'}_k$.

Go to stage $s+1$.

End stage s

endif

End { procedure for enumerating $W^{\varphi'}_k$ }

Clearly φ' is an acceptable programming system since φ-indices can be reduced to φ'-indices by the recursive function $\lambda y.[(n+1)y]$. Also, let w be a monotone increasing recursive function that reduces φ'-indices to φ-indices. We now describe a program for learning function $\mathbf{F}'$.

Begin $\mathbf{F}'(T[x])$

Let $P_x = \{j \leq x \mid [\mathbf{F}(T[x]) \in S^x_j] \ \wedge [\mathrm{content}(T[\mathbf{PreviousMindChange}(j,x)]) \subseteq W^{\varphi}_{j,x}] \wedge [W^{\varphi}_{j,\mathbf{PreviousMindChange}(j,x)} \subseteq \mathrm{content}(T[x])]\}$.

if $P_x = \emptyset$

then

Output 0

else

Let $k_x = \min(P_x)$.

Output $(n+1)k_x + r_x$, where $1 \leq r_x \leq n$ and $G_{k_x}^{x,r_x} = \mathbf{F}(T[x])$.

endif

End $\mathbf{F}'(T[x])$

We assert that, for any text T for $L \in \mathcal{L}$, for all but finitely many x, $\mathbf{F}'$ on $T[x]$ will output a grammar of the form $(n+1)k_x + r_x$, where $1 \leq r_x \leq n$, such that the following two conditions hold.

(i) $k_x \leq \text{mingrammar}_{\varphi}(L)$; and
(ii) $W^{\varphi'}_{(n+1)k_x+r_x} =^* L$.

Moreover, $\mathbf{F}'(T)\Downarrow_n$. If our assertion is true, then the theorem follows since, for all but finitely many x, $(n+1)k_x + r_x \leq h(\text{mingrammar}_{\varphi'}(L))$, where $h = \lambda y.[(n+1)(w(y)+1)]$. This is because $k_x \leq \text{mingrammar}_{\varphi}(L)$ implies that $(n+1)k_x + r_x \leq (n+1)(\text{mingrammar}_{\varphi}(L)+1)$ $\leq (n+1)(w(\text{mingrammar}_{\varphi'}(L))+1) = h(\text{mingrammar}_{\varphi'}(L))$. It remains to prove our assertion.

Let $L \in \mathcal{L}$ and T be any text for L. Consider $\mathbf{F}'$ on T. Below let P_x, k_x and r_x be as in $\mathbf{F}'(T[x])$.

Claim 26 $(\forall^\infty x)[\text{mingrammar}_{\varphi}(L) \in P_x]$.

Proof of Claim 26. Since $\mathbf{F}$ is n-ary order independent and $\mathbf{F}$ $\mathbf{TxtFex}^*_n$-identifies L, $(\exists D \mid \text{card}(D) \leq n)(\forall \text{ texts } T \text{ for } L)[\mathbf{F}(T)\Downarrow_n = D]$. Clearly, for all but finitely many x, $D \subseteq S^x_{\text{mingrammar}_{\varphi}(L)}$. Hence, for all but finitely many x, $\text{mingrammar}_{\varphi}(L) \in P_x$. ∎ (Claim 26)

Claim 27 $(\forall j)[(\exists^\infty x)[j \in P_x] \Rightarrow (\exists D \mid \text{card}(D) \leq n)[\mathbf{F}(T_j)\Downarrow_n = D]]$.

Proof of Claim 27. We have two cases.

Case 1: $W^{\varphi}_j = L$.

Since $\mathbf{F}$ is n-ary order independent and $\mathbf{F}$ $\mathbf{TxtFex}^*_n$-identifies L, clearly there exists a set D of cardinality at most n such that $\mathbf{F}(T_j)\Downarrow_n = D$.

Case 2: $W^{\varphi}_j \neq L$.

Suppose $\neg[(\exists D \mid \text{card}(D) \leq n)[\mathbf{F}(T_j)\Downarrow_n = D]]$. Let m be such that, $(\text{content}(T[m]) \not\subseteq W^{\varphi}_j) \vee (W^{\varphi}_{j,m} \not\subseteq L)$. Clearly such an m exists since $W^{\varphi}_j \neq L$. Also, since $\neg(\exists D \mid \text{card}(D) \leq n)[\mathbf{F}(T_j)\Downarrow_n = D]$, $\lim_{x\to\infty} \mathbf{PreviousMindChange}(j,x)$ is ∞. Hence, $(\forall x \mid \mathbf{PreviousMindChange}(j,x) > m)[j \notin P_x]$. ∎ (Claim 27)

Claim 28 $(\forall j)(\forall r \mid 1 \leq r \leq n)[\lim_{x\to\infty} G_j^{x,r}\downarrow \Rightarrow W^{\varphi'}_{(n+1)j+r} =^* W^{\varphi}_{\lim_{x\to\infty} G_j^{x,r}}]$.

Proof of Claim 28. Clear from the construction of φ'. ∎ (Claim 28)

Let $S = \{k \mid (\exists^\infty x)[\mathbf{F}(T[x]) = k]\}$. From Claim 26 and the description of $\mathbf{F}'$, we have that, for all but finitely many x, $k_x \leq \text{mingrammar}_\varphi(L)$. For each $l \in S$, let $p_l = \min(\{j \mid (\exists^\infty x)[j \in P_x \wedge l \in S_j^x]\})$. From Claim 27, we have that, for each $l \in S$, for all but finitely many x, if $\mathbf{F}(T[x]) = l$, then $p_l \in P_x$. Also, for each $l \in S$, let r_l be such that, for all but finitely many x, $G_{p_l}^{x,r_l} = l$ (that such an r_l exists follows from Claim 27). Thus, it follows that, for all but finitely many x, $\mathbf{F}(T[x]) = l \Rightarrow \mathbf{F}'(T[x]) = (n+1)p_l + r_l$. Also, by Claim 28, $W^{\varphi'}_{(n+1)p_l+r_l} =^* W_l^{\varphi}$. The theorem follows. ∎ (Theorem 25)

Our next theorem (Theorem 29) contrasts sharply and surprisingly with both Theorem 25 just above and the situation regarding the learning of programs for functions [13, 14]. Theorem 29 says that, if we allow an unbounded finite number of anomalies in final grammars and if the number of different grammars converged to in the limit is *unbounded* but finite, then there *is* a loss of learning power in requiring that the final grammars be nearly minimal size.

Theorem 29 $(\mathbf{TxtFex}^0_* - \mathbf{TxtMfex}^*_*) \neq \emptyset$.

Our proof of Theorem 29 employs Case's operator recursion theorem [4], an infinitary recursion theorem. The construction uses only a finite number of self-other referential grammars, but the finite number is not determined in advance as it is with finitary multiple recursion theorems.

Proof of Theorem 29. Let $\mathcal{L} = \{L \in \mathcal{E} \mid L \text{ is infinite } \wedge [\text{card}(\{y \mid (\exists x)[\langle x, y\rangle \in L]\}) < \infty] \wedge (\forall^\infty \langle x, y\rangle)[\langle x, y\rangle \in L \Rightarrow W_y = L]\}$. Clearly $\mathcal{L} \in \mathbf{TxtFex}^0_*$. Suppose by way of contradiction, there exists a learning function $\mathbf{F}$ which $\mathbf{TxtMfex}^*_*$-identifies $\mathcal{L}$. Let h be as in the Definition 10 for $\mathbf{TxtMfex}^*_*$-identification of $\mathcal{L}$ by $\mathbf{F}$. By Remark 11 we may assume, without loss of generality, that h is monotone increasing.

By implicit use of the operator recursion theorem, there exists a recursive, one-to-one function p such that, for $i \leq h(p(0)) + 1$, $W_{p(i)}$ may be described as follows. We now proceed to give an informal effective construction of the $W_{p(i)}$'s in successive stages $s \geq 0$. Let $\sigma_0 = \emptyset$. Go to stage 0.

Begin stage s

Let $i = s$.

repeat

for index := 0 **to** $h(p(0))+1$ **do**

Enumerate $\langle i, p(\text{index})\rangle$ into $W_{p(\text{index})}$.

endfor

$i := i+1$.

until $(\exists r \leq h(p(0))+1)[\mathbf{F}(\sigma_s \diamond (\langle s, p(r)\rangle) \diamond (\langle s+1, p(r)\rangle) \diamond \cdots \diamond (\langle i-1, p(r)\rangle)) > h(p(0))]$;

(∗ **until** *clause succeeds ∗)*

for index := 0 **to** $h(p(0))+1$ **do**

Enumerate $\langle s, p(\text{index})\rangle, \langle s+1, p(\text{index})\rangle, \cdots, \langle i-1, p(\text{index})\rangle$ into $W_{p(0)}, W_{p(1)}, \cdots, W_{p(h(p(0))+1)}$

endfor

Let r be as found in the **until** clause above.

Let σ_{s+1} be an extension of $\sigma_s \diamond (\langle s, p(r)\rangle) \diamond (\langle s+1, p(r)\rangle) \diamond \cdots \diamond (\langle i-1, p(r)\rangle)$ such that content$(\sigma_{s+1}) = W_{p(0)}$ enumerated till now.

Go to stage $s+1$.

End stage s

We have the following two cases.

Case 1: Each stage terminates.

Then $W_{p(0)} = W_{p(1)} = \cdots = W_{p(h(p(0))+1)}$. Let $L = W_{p(0)}$. Clearly $L \in \mathcal{L}$. Also, $T = \bigcup_{s\in N} \sigma_s$ is a text for L and $\mathbf{F}$ on T outputs a grammar greater than $h(p(0))$ infinitely often. Since h is monotone increasing, $\mathbf{F}$ on T outputs a grammar greater than $h(\text{mingrammar}(L))$ infinitely often. Thus, $\mathbf{F}$ does not $\mathbf{TxtMfex}^*_*$-identify $\mathcal{L}$.

Case 2: Some stage s starts but never terminates.

For $0 \leq r \leq h(p(0))+1$, let $L_r = W_{p(r)}$. Clearly, for $0 \leq r \leq h(p(0))+1$, $L_r \in \mathcal{L}$. Also, for $0 \leq r < q \leq h(p(0))+1$, $L_r \neq^* L_q$. Furthermore, $(\forall r \leq h(p(0))+1)(\exists$ a text T for $L_r)(\forall^\infty x)[\mathbf{F}(T[x]) \leq h(p(0))]$. If this were not the case, then the **until** clause in stage s would have succeeded and stage s would have terminated. Since the class of languages $\{L_r \mid 0 \leq r \leq h(p(0))+1\}$ has $h(p(0))+2$ languages, pairwise infinitely different, and the set of grammars $\{i \mid 0 \leq h(p(0))\}$ has $h(p(0))+1$ grammars, there exists an r such that $(\forall i \leq h(p(0)))[L_r \neq^* W_i]$. Thus $\mathbf{F}$ does not $\mathbf{TxtMfex}^*_*$-identify $L_r \in \mathcal{L}$.

From the above two cases, it follows that $\mathcal{L} \notin \mathbf{TxtMfex}^*_*$. ∎

We generalize recursion theorem arguments from [22, 13, 14, 8] to show Theorem 30 immediately below.

Theorem 30 $(\mathbf{TxtFex}_1^0 - \mathbf{TxtMfex}_*^m) \neq \emptyset$.

Corollary 31 just below follows from Theorems 25 and 30 above.

Corollary 31 $\mathbf{TxtMfex}_n^m \subset \mathbf{TxtFex}_n^m \subset \mathbf{TxtMfex}_n^*$.

Proof of Theorem 30. For any recursive function f, define $L_f = \{\langle x, y\rangle \mid f(x) = y\}$. Let $\mathcal{C} = \{f \mid (\forall^\infty x)[f(x) = 0]\}$. Let $\mathcal{L} = \{L_f \mid f \in \mathcal{C}\}$. It is easy to see that $\mathcal{L} \in \mathbf{TxtFex}_1^0$. Also the proof of $\mathcal{C} \notin \mathbf{Mfex}_*^m$ (in [14]) can be easily modified to show that $\mathcal{L} \notin \mathbf{TxtMfex}_*^m$. ∎

In summary: there is *no* loss of learning power in demanding that the final grammars be nearly minimal size *iff* one is willing to tolerate an *unbounded*, finite number of anomalies in the final grammars *and* there is a *constant* bound on the number of different grammars converged to in the limit. Hence, if we allow an unbounded, finite number of anomalies in the final grammars *and* the number of different grammars converged to in the limit is *unbounded* but finite or if there is a *constant* bound on the number of anomalies allowed in the final grammars, then there *is* a loss of learning power in requiring that the final grammars be nearly minimal size.

6 Relation to Behaviorally Correct Text Identification

We now extend our definitions and results to behaviorally correct identification [11].

Definition 32

(a) **F TxtBc^a-*identifies* L** (written: $L \in \mathbf{TxtBc}^a(\mathbf{F})$) $\iff$
($\forall$ texts T for L)$(\forall^\infty n)[W_{\mathbf{F}(T[n])} =^a L]$.

(b) $\mathbf{TxtBc}^a = \{\mathcal{L} \mid (\exists \mathbf{F})[\mathcal{L} \subseteq \mathbf{TxtBc}^a(\mathbf{F})]\}$.

Definition 32 is from [11]. The $a \in \{0, *\}$ cases were independently introduced in [28, 27].

We sometimes write **TxtBc** for $\mathbf{TxtBc}^0$.

We now extend the above definition to nearly minimal identification.

Definition 33

(a) **F TxtMbc^a-*identifies* $\mathcal{L}$** (written $\mathcal{L} \subseteq \mathbf{TxtMbc}(\mathbf{F})$) $\iff$ $\mathcal{L} \subseteq \mathbf{TxtBc}^a(\mathbf{F})$ and $(\exists h \in \mathcal{R})(\forall L \in \mathcal{L})(\forall$ texts T for $L)(\forall^\infty n)[\mathbf{F}(T[n]) \leq h(\text{mingrammar}(L))]$.

(b) $\mathbf{TxtMbc}^a = \{\mathcal{L} \mid (\exists \mathbf{F})[\mathcal{L} \subseteq \mathbf{TxtMbc}^a(\mathbf{F})]\}$.

Since, for all L and h, there exist only finitely many grammars $\leq h(\text{mingrammar}(L))$, we immediately have

Proposition 34 *For all a, $\mathbf{TxtMbc}^a = \mathbf{TxtMfex}^a_*$.*

We now state a result from [11] which is useful in proving Theorem 36.

Theorem 35 [11] *For all m, $\{L \mid \text{card}(\overline{L}) \leq 2m+1\} \notin \mathbf{TxtBc}^m$.*

Theorem 36 *For all m, $\mathbf{TxtMfex}^{2m+1}_1 - \mathbf{TxtBc}^m \neq \emptyset$.*

Proof of Theorem 36. Let $\mathcal{L}_m = \{L \mid \text{card}(\overline{L}) \leq 2m+1\}$. Let i_N be a grammar for N. Let $\mathbf{F}$ be such that for all σ, $\mathbf{F}(\sigma) = i_N$. Clearly, $\mathbf{F}$ $\mathbf{TxtMfex}^{2m+1}_1$-identifies $\mathcal{L}_m$. By Theorem 35, $\mathcal{L}_m \notin \mathbf{TxtBc}^m$. ∎

The above, along with Theorems 37 and 38, gives the relationship between all the criteria, discussed in this paper, involving nearly minimal size grammars and the criteria $\mathbf{TxtBc}^a$.

Theorem 37 [6] *For all m, $\mathbf{TxtFex}^{2m}_* \subseteq \mathbf{TxtBc}^m$.*

Theorem 38 [6] $\mathbf{TxtBc} - \mathbf{TxtFex}^*_* \neq \emptyset$.

7 Conclusion

The present paper investigated the impact of requiring the final grammars to be nearly minimal in the context of vacillatory identification of languages. A useful characterization of this size restricted notion was established and employed to show that there are collections of languages that can be learned by converging in the limit to up to $n+1$ correct grammars, but that cannot be learned by converging to up to n unrestricted grammars even if these latter grammars are allowed to have a finite number of anomalies per grammar. In the terminology of the present paper, there are collections of languages that can be $\mathbf{TxtMfex}^0_{n+1}$-identified, but that cannot be $\mathbf{TxtFex}^*_n$-identified.

It was also shown that there is no loss of learning power in demanding that the final grammars be nearly minimal iff one is willing to tolerate an unbounded, finite number of anomalies in the final grammars and there is a constant bound on the number of different grammars converged to in the limit. That is, for $n \in N^+$, $\mathbf{TxtMfex}^*_n = \mathbf{TxtFex}^*_n$. It was also shown that this latter result does not hold for $n = *$ by establishing $\mathbf{TxtMfex}^*_*$ to be properly contained in $\mathbf{TxtFex}^*_*$.

Figure 1 provides a summary of the results discussed in the present paper.

8 Acknowledgement

We would like to thank an anonymous referee for several valuable comments. The research was supported in part by NSF grants CCR 8320136 at the University of Rochester and CCR 8713846 at SUNY, Buffalo. Work done in part while the second author was at the University of Rochester and the first and third authors were at SUNY, Buffalo. The preliminary version of this paper appeared in the Proceedings of the Second Annual Workshop on Computational Learning Theory, Santa Cruz, California, August 1989. [9]

References

[1] J. M. Barzdin and K. Podnieks. The theory of inductive inference. In *Mathematical Foundations of Computer Science*, 1973.

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

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

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

[5] 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 Publishers, Inc., 1988. Expanded in [7].

[6] J. Case. The power of vacillation in language learning. In preparation, 1992.

[7] J. Case. The power of vacillation in language learning. Technical Report 93-08, University of Delaware, 1992. Expands on [5]; journal article under review.

[8] J. Case and H. Chi. Machine learning of nearly minimal size grammars. Unpublished Manuscript, 1986.

[9] J. Case, S. Jain, and A. Sharma. Convergence to nearly minimal size grammars by vacillating learning machines. In R. Rivest, D. Haussler, and M. K. Warmuth, editors, *Proceedings of the Second Annual Workshop on Computational Learning Theory, Santa Cruz, California*, pages 189–199. Morgan Kaufmann Publishers, Inc., August 1989.

[10] J. Case, S. Jain, and A. Sharma. Complexity issues for vacillatory function identification. In *Proceedings, Foundations of Software Technology and Theoretical Computer Science, Eleventh Conference, New Delhi, India. Lecture Notes in Computer Science 560*, pages 121–140. Springer-Verlag, December 1991.

[11] 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, pages 107–115. Springer-Verlag, Berlin, 1982.

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

[13] K. Chen. *Tradeoffs in Machine Inductive Inference*. PhD thesis, SUNY at Buffalo, 1981.

[14] K. Chen. Tradeoffs in inductive inference of nearly minimal sized programs. *Information and Control*, 52:68–86, 1982.

[15] R. Freivalds. Minimal Gödel numbers and their identification in the limit. *Lecture Notes in Computer Science*, 32:219–225, 1975.

[16] M. Fulk. *A Study of Inductive Inference machines*. PhD thesis, SUNY at Buffalo, 1985.

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

[18] E. M. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[19] J. Hopcroft and J. Ullman. *Introduction to Automata Theory Languages and Computation*. Addison-Wesley Publishing Company, 1979.

[20] S. Jain and A. Sharma. Restrictions on grammar size in language identification. In David Powers and Larry Reeker, editors, *Proceedings MLNLO'91, Machine Learning of Natural Language and Ontology, Stanford University, California. Document* **D91-09***, DFKI: Kaiserslautern FRG, 1991*, pages 87–92, March 1991.

[21] S. Jain and A. Sharma. Program size restrictions in computational learning. *Theoretical Computer Science A*, 1994. To appear.

[22] E. B. Kinber. On a theory of inductive inference. *Lecture Notes in Computer Science*, 56:435–440, 1977.

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

[24] D. Osherson, M. Stob, and S. Weinstein. Note on a central lemma of learning theory. *Journal of Mathematical Psychology*, 27:86–92, 1983.

[25] D. Osherson, M. Stob, and S. Weinstein. Learning theory and natural language. *Cognition*, 17:1–28, 1984.

[26] D. Osherson, M. Stob, and S. Weinstein. *Systems that Learn, An Introduction to Learning Theory for Cognitive and Computer Scientists*. MIT Press, Cambridge, Mass., 1986.

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

[28] D. Osherson and S. Weinstein. A note on formal learning theory. *Cognition*, 11:77–88, 1982.

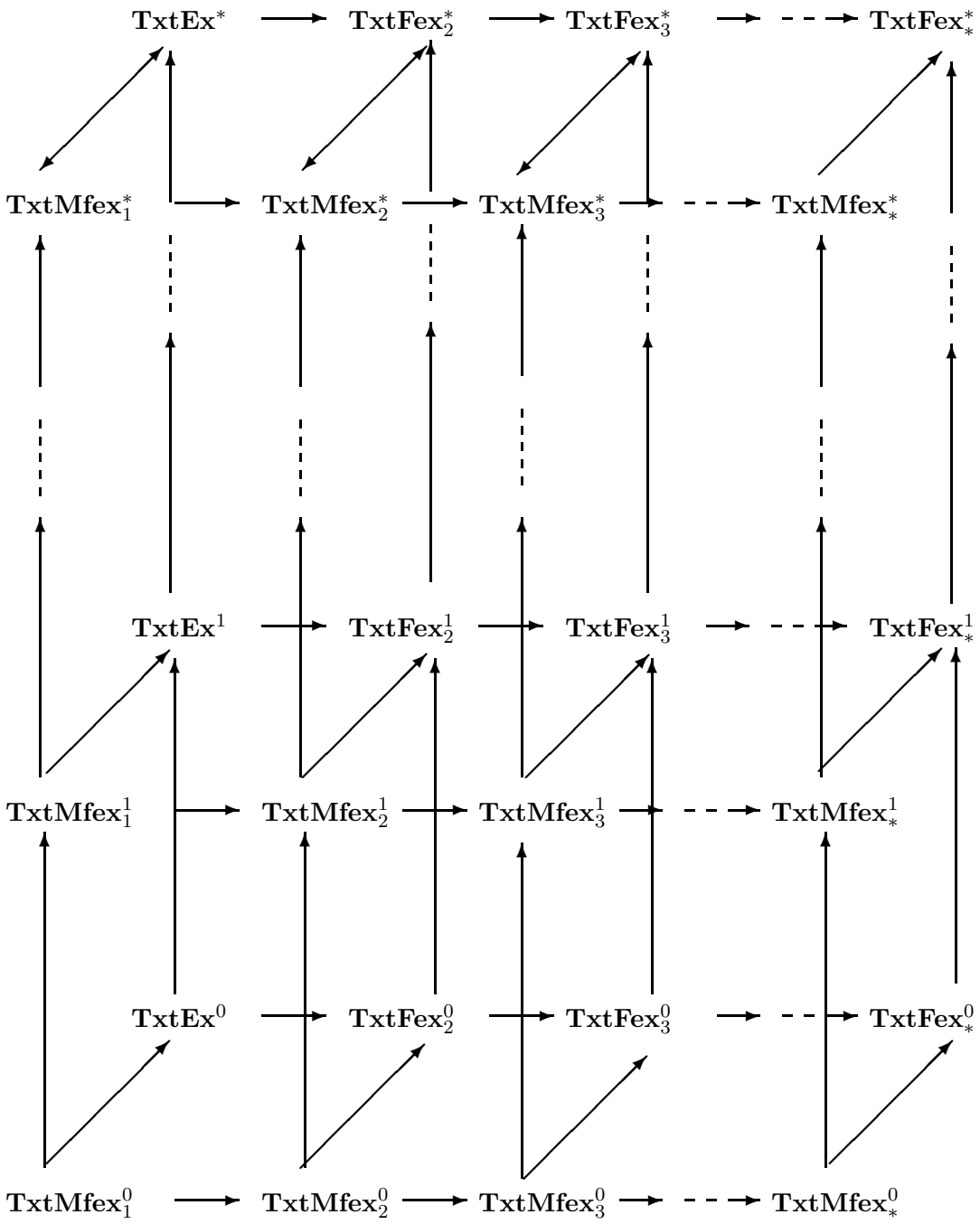
[29] S. Pinker. Formal models of language learning. *Cognition*, 7:217–283, 1979.

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

[31] J. Royer and J. Case. *Intensional Subrecursion and Complexity Theory*. Research Notes in Theoretical Science, Pitman Press, 1992. Under preparation.

[32] K. Wexler. On extensional learnability. *Cognition*, 11:89–95, 1982.

[33] K. Wexler and P. Culicover. *Formal Principles of Language Acquisition*. MIT Press, Cambridge, Mass, 1980.



$A \longrightarrow B$ denotes $A \subset B$.

$A \longleftrightarrow B$ denotes $A = B$.

Figure 1. Summary of results