# The Power of Vacillation in Language Learning

John Case*
Department of Computer and Information Sciences
University of Delaware
Newark, DE 19716, USA

**Abstract**

Some extensions are considered of Gold's influential model of language learning by machine from positive data. Studied are criteria of successful learning featuring convergence in the limit to vacillation between several alternative correct grammars. The main theorem of this paper is that there are classes of languages that can be learned if convergence in the limit to up to $(n+1)$ exactly correct grammars is allowed but which cannot be learned if convergence in the limit is to no more than $n$ grammars, where the no more than $n$ grammars can each make finitely many mistakes. This contrasts sharply with results of Barzdin and Podnieks and, later, Case and Smith, for learnability from both positive and negative data.

A subset principle from a 1980 paper of Angluin is extended to the vacillatory and other criteria of this paper. This principle, provides a necessary condition for circumventing overgeneralization in learning from positive data. It is applied to prove another theorem to the effect that one can optimally eliminate $\frac{1}{2}$ the mistakes from final programs for vacillatory criteria if one is willing to converge in the limit to infinitely many different programs instead.

Child language learning may be sensitive to the order or timing of data presentation. It is shown, though, that, for the vacillatory success criteria of this paper, there is no loss of learning power for machines which are *in*sensitive to order in several ways simultaneously. For example, *partly set-driven* machines attend only to the *set* and *length of sequence* of positive data, not the actual sequence itself. A machine $\mathbf{M}$ is *weakly $n$-ary order independent* $\stackrel{\text{def}}{\Leftrightarrow}$ for each language $L$ on which, for some ordering of the positive data about $L$, $\mathbf{M}$ converges in the limit to a finite set of grammars, there is a finite set of grammars $D$ (of cardinality $\leq n$) such that $\mathbf{M}$ converges to a subset of this same $D$ for *each* ordering of the positive data for $L$. The most difficult to prove theorem in the paper implies that machines which are simultaneously partly set-driven and weakly $n$-ary order independent do not lose learning power for converging in the limit to up to $n$ grammars. Several variants of this theorem are obtained by modifying its proof, and some of these variants have application in this and other papers. Along the way it is also shown, for the vacillatory criteria, that learning power is not increased if one restricts the sequence of positive data presentation to be computable. Some of these results are non-trivial lifts of prior work for the $n = 1$ case due to the Blums; Wiehagen; Osherson, Stob and Weinstein; Schäfer; and Fulk.

*Email: 'case@cis.udel.edu'. URL: 'http://www.eecis.udel.edu/~case'. The research was supported in part by NSF grant # CCR-8713846. The author thanks the University of Rochester's Computer Science Department for support and for providing an excellent working environment academic year 1987-8 during which some of the work on the present paper was completed. The author is also grateful to anonymous referees of a previous draft and others mentioned in the text for many helpful comments. This paper is an expansion with corrections of both the conference article [Cas88] and the subsequent technical report [Cas92a], and a variant is being considered for possible journal publication.

# 1 Introduction

Gold in [Gol67] introduced his seminal model of language learning: Imagine, as pictured in (1) just below, a machine $\mathbf{M}$ being fed data about membership in a (formal) language $L$ and, as a result, outputting over time a series of grammars $p_0, p_1, p_2, \ldots, p_t, p_{t+1}, \ldots$ *conjectured* to be for $L$.

$$p_0, p_1, p_2, \ldots, p_t, p_{t+1}, \ldots \leftarrow \mathbf{M} \leftarrow \text{data re } L \tag{1}$$

For our present purposes, it will suffice to consider two kinds of data presentation and one kind of success from [Gol67].

Data about $L$ is *either*

1. *informant*, a listing of every possible language element with an clear indication of whether or not it is in $L$ *or*

2. *text*, an arbitrary listing of all and only the elements of $L$.

Gold took quite seriously, as a model of child language learning, the case of data presentation by arbitrary text, the case where $\mathbf{M}$ receives all and only *positive* information about $L$. Justification for this point of view can be found, for example, in [BH70, Bra71], where it is noted from field work that children don't need corrections to learn language.

Re successful language learning, referring to (1) above: for Gold, machine $\mathbf{M}$ *identifies* language $L \stackrel{\text{def}}{\Leftrightarrow} \mathbf{M}$ fed any text for $L$, outputs a corresponding sequence $p_0, p_1, p_2, \ldots$ such that, for some $t$, $p_t = p_{t+1} = p_{t+2} = \ldots$ and $p_t$ is a correct grammar for $L$. In other words, $\mathbf{M}$ *identifies* $L \Leftrightarrow$ on each text for $L$, the corresponding conjectures of $\mathbf{M}$ converge, in the limit, to some fixed final conjecture and that final conjecture is correct.[1] Gold showed that *no* $\mathbf{M}$ so identifies the entire class of regular languages [HU79], but *some* $\mathbf{M}$ *does* identify the class of finite languages. Angluin [Ang80, Ang82] presents other classes $\mathcal{L}$ natural from the perspective of formal language theory such that some $\mathbf{M}$ identifies each language in $\mathcal{L}$.

Many *cognitive scientists* seek to model all of cognition by computer program [Pyl84, JL88], and Gold's model of language learning from text (positive information) by machine has been very influential in contemporary theories of natural language and in mathematical work explicitly motivated by its possible connection to human language learning (see, for example, [Pin79, WC80, Wex82, OSW82, OSW84, Ber85, Gle86, Cas86, OSW86a, OSW86b, Ful85, Ful90a, Kir92, BCJ95]).

In the present paper we consider some new criteria of success extending Gold's basic model above. Suppose we fix an integer $n > 0$. Consider the following criterion of success (again based on (1) above). We say that $\mathbf{M}$ $\mathbf{TxtFex}_n$-*identifies* $L \stackrel{\text{def}}{\Leftrightarrow} \mathbf{M}$, on any text for $L$, outputs corresponding conjectures $p_0, p_1, p_2, \ldots$ such that there is a $t$ for which

1. the sequence $p_t, p_{t+1}, p_{t+2}, \ldots$ contains at most $n$ distinct grammars *and*

2. each of the grammars $p_t, p_{t+1}, p_{t+2}, \ldots$ is correct.

Of course Gold's identification criterion above is just $\mathbf{TxtFex}_1$-identification. It is well known [Rog67] that equivalent grammars (e.g., $p_t, p_{t+1}, p_{t+2}, \ldots$ as above) can be so different from one another that in some cases it is not possible to prove in Zermelo-Frankel Set Theory [Jec78] that they are equivalent. This suggests that a suitably clever $\mathbf{M}$ might be able to $\mathbf{TxtFex}_{n+1}$-identify a

[1] N.B. It is importantly not required that $\mathbf{M}$ signal when it has reached its final conjecture — in general it doesn't know when and if it has.

larger class of languages than any machine, however clever, could $\mathbf{TxtFex}_n$-identify. Unfortunately, it was already known [BP73] that, at least in the case where the data is informant instead of text, one gets no more learning power with $(n+1)$ correct programs in the limit than with $n$. Surprisingly, then, the main theorem of the present paper (Theorem 1 in Section 3 below) implies that, nonetheless, for learning from *text*, larger classes of languages *can* be learned with up to $(n+1)$ correct programs in the limit than with up to $n$.

Theorem 1 suggests, then, the *possibility* that evolutionary pressure for increased learning power may have resulted in human language learning strategies that involve convergence to *vacillating* between $n > 1$ correct grammars in the limit. This is examined more critically in Section 7 below. Regarding, though, the size of $n$, we note that at least one of $n$ distinct grammars would have to be of size proportional to the size of $n$ (i.e., to $\log n$); hence, for extraordinarily *large* $n$, at least one of $n$ distinct grammars would be too large to fit in our heads — unless, as seems unlikely, human storage mechanisms admit infinite regress. Osherson and Weinstein [OW82a] introduced the case where the number of final grammars is finite, but *unbounded*, and [CL82, OW82a] independently (see also [OW82b]) introduced the case where the number of final grammars is infinite (**TxtBc**-identification). We briefly introduced the case, discussed above, of up to $n$ final grammars in [Cas86].

The proof of Theorem 1 employs an $(n+1)$-ary self-reference argument [Cas94], and an informal thesis is presented and discussed after the statement of Theorem 1 that self-referential examples witnessing an existence theorem presage natural examples witnessing that theorem.

[CL82] considered, among other things, the learning of grammars for languages where a *single* final grammar is allowed to have a bounded number of mistakes (*anomalies*). The mistakes are about which objects are (and which are not) in the corresponding language. In [CS78, CS83, Cas86] there are discussion, motivation and interpretation of results about inferring anomalous programs for functions. The results in [CS78, CS83, Cas86] and in this paper show that allowing anomalies increases learning power. Clearly anomalous programs are tolerable provided the number of anomalies is *small*. Hence, it is plausible that people have evolved language learning strategies that exploit the greater learning power achieved by converging to slightly incorrect grammars. Theorem 1 says, more generally than indicated above, that, for each $n > 0$, some classes of languages can be algorithmically learned (in the limit) by converging to *up to* $n+1$ different, exactly correct grammars; *but* these classes can*not* be learned by converging (in the limit) to up to $n$ different grammars, where the up to $n$ grammars are each allowed to have a finite number of anomalies

Corollary 4 in Section 3 below specifies a two-dimensional hierarchy involving $\mathbf{TxtFex}_b^a$-identification: learning up to $b$ final grammars each with up to $a$ mistakes.

Theorem 2 in Section 3 implies that, in passing from learning finitely many anomalous grammars in the limit to learning infinitely many, one can eliminate $\frac{1}{2}$ the anomalies, and that that's optimal! Intuitively, since, with positive data only, one is missing $\approx \frac{1}{2}$ the information, one can eliminate $\frac{1}{2}$ the anomalies only.

If $L$ is a non-empty language, then *some* texts for $L$ are *non*-computable sequences, but in a completely computable universe, no parents can generate a non-computable sequence of data for their children. Hence, it is interesting to consider $\mathbf{RecTxtFex}_b^a$*-identification*, $\mathbf{TxtFex}_b^a$-identification except that success is only required on *computable* presentations of positive data, on all *recursive* texts. It might be expected that a suitably clever machine $\mathbf{M}$ might be able to exploit the recursiveness of texts to learn larger classes of languages than any machine required to succeed on arbitrary texts, but Corollary 1 in Section 3 below implies that this is not the case (generalizing the $b = 1$ case essentially from [Wie77, BB75]). We say, then, that *the restriction to recursive texts is circumvented.*

Angluin, in her seminal paper [Ang80], presents a severe constraint on $\mathbf{TxtFex}_1$-identification of classes of languages: the *subset principle.* Basically she shows that, if $\mathbf{M}$ $\mathbf{TxtFex}_1$-identifies a class of languages $\mathcal{L}$, then, for each $L \in \mathcal{L}$, there is a *finite* set $D$ (called a *tell tale*) contained in $L$ such that $D$ is not contained in any proper sublanguage of $L$ in $\mathcal{L}$. Intuitively, this necessary condition circumvents overgeneralization in learning from positive data [Ang80, Ber85]. Theorem 3 in Section 4 below generalizes the subset principle to the criteria of success $\mathbf{TxtFex}_b^a$-identification and $\mathbf{TxtBc}^a$-identification, where the $a$ in $\mathbf{TxtBc}^a$-identification allows each of the infinitely many final grammars converged to to have up to $a$ anomalies. Theorem 3 is also used to prove Theorem 2 in Section 3 below.

A child learning a language may or may not be sensitive to *the order or timing of presentation of positive data.* For $\mathbf{TxtFex}_b^a$-identification (and variants thereof) we mathematically consider several kinds of *in*sensitivity of a machine $\mathbf{M}$ to data order:

1. *set-driven*: $\mathbf{M}$'s output at any point depends only on the *set* of positive data it's seen up to that point (not on the sequence in which it was presented).

2. *partly set-driven*: $\mathbf{M}$'s output at any point depends only on the *set* of positive data it's seen up to that point *and on the length* of the sequence in which it was presented.

3. *b-ary order independent*: for languages $L$ on which for some text $\mathbf{M}$ converges to a finite set of final grammars, $\mathbf{M}$ converges to the *same* set (of cardinality $\leq b$) of final grammars for each text for $L$.

4. *weakly b-ary order independent*: for languages $L$ on which for some text $\mathbf{M}$ converges to a finite set of final grammars, there is a finite set of grammars $D$ (of cardinality $\leq b$) such that $\mathbf{M}$ converges to a subset of this $D$ for each text for $L$.

In Section 5 below, we prove several theorems each witnessing that, for suitably clever $\mathbf{M}$'s *simultaneously* exhibiting some insensitivities as above and circumventing the restriction to recursive texts, there is *no* loss of learning power (with respect to $\mathbf{TxtFex}_b^a$-identification or the variants thereof). For *example*, Theorem 4 (in Section 5) implies that the power of $\mathbf{TxtFex}_b^a$-identification is unaffected by the restriction to $\mathbf{M}$'s which are simultaneously partly set-driven and weakly $b$-ary order independent and which circumvent the restriction to recursive texts. Theorem 4 is the hardest theorem herein to prove, and the other theorems in Section 5 are proved by modifications, and/or simplifications of the proof of Theorem 4. Some of the theorems in Section 5 generalize predecessors for $\mathbf{TxtFex}_1^0$-identification [BB75, WC80, SR84, Ful85, OSW86b, Ful90a] but are much harder to prove. Some of the theorems in Section 5 are applied in the present paper and in other papers.

In Section 7 we discuss briefly computable universe hypotheses, present some critical discussion as promised above, and sketch some areas for future investigation.

## 2 Preliminaries

We now proceed more formally.

$\mathbf{N}$ denotes the set of *natural numbers*, $\{0, 1, 2, \dots\}$.

$\varphi$ denotes a fixed *acceptable* programming system for the partial computable functions: $\mathbf{N} \to \mathbf{N}$ [Rog58, MY78, Ric80, Ric81, Roy87]. $\varphi_p$ denotes the partial computable function computed by the program (with code number) $p$ in the $\varphi$-system.[2] Thanks to the device of Gödel or code numbering

[2]The *acceptable* systems are those universal programming systems such as Turing machines, C, and Lisp into which one can compile from any programming system. We characterized them as those universal systems for the partial computable functions in which one can implement any control structure [Roy87].

[Rog67] we can treat languages over any finite alphabet as subsets of $\mathbf{N}$. $W_p \stackrel{\text{def}}{=}$ the domain of $\varphi_p$, the r.e. language ($\subseteq \mathbf{N}$) recognized (or enumerated) by program (grammar) $p$ in the $\varphi$-system [Rog67].

**Definition 1** *A* language learning function *is a computable mapping from finite sequences, of natural numbers and #'s, into (Gödel numbers of) programs (grammars) in the $\varphi$-system.*

$\mathcal{E}$ denotes the class of all r.e. languages ($\subseteq \mathbf{N}$).

**Definition 2** *A* text for *a language $L$ is a mapping $T$ from* $\mathbf{N}$ *into* $(\mathbf{N} \cup \{\#\})$ *such that $L$ is the set of natural numbers in the range of $T$. $T$ is said to be* for $L \Leftrightarrow$ *$T$ is a text for $L$. The* content of *a sequence, of natural numbers and #'s, is the set of natural numbers in its range.* content$(\cdot)$ *denotes the content of its argument.*

Intuitively, one can think of a text for a language as an enumeration of the objects in the language with the #'s representing pauses in the listing of such objects. For example, the only text for the empty language is just an infinite sequence of #'s.

Intuitively, if $\mathbf{F}$ is a learning function and $\sigma$ is a finite initial segment of a text for a language $L$, then $\mathbf{F}(\sigma)$ represents $\mathbf{F}$'s conjecture as to a grammar for $L$ based on the data about $L$ in $\sigma$.

Variables $\sigma$ and $\tau$ (with or without decorations[3]) range over finite initial segments of texts $T$. $\|\sigma\|$ denotes the length of $\sigma$. $\sigma \diamond \sigma'$ denotes the sequence formed by adding $\sigma'$ to the end of $\sigma$. Hence, if $\tau = \sigma \diamond \sigma'$, then, for all $x \in \mathbf{N}$,

$$\tau(x) = \begin{cases} \sigma(x), & \text{if } x < \|\sigma\|; \\ \sigma'(x - \|\sigma\|), & \text{if } \|\sigma\| \leq x < \|\sigma\| + \|\sigma'\|; \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

Furthermore, $\sigma \diamond x$, where $x \in (\mathbf{N} \cup \{\#\})$, denotes $\sigma \diamond \sigma'$, where $\sigma' = \{(0, x)\}$.

card$(D)$ denotes the cardinality of $D$. $\mathbf{I}^+$ denotes the set of positive integers. We take $a$ and $c$ to range over $(\mathbf{N} \cup \{*\})$ and $b$ and $d$ to range over $(\mathbf{I}^+ \cup \{*\})$. Intuitively, $*$ denotes the unbounded, but finite. For example, 'card$(D) \leq *$' means that $D$ is finite. We adopt the convention that $(\forall i \in N)[i < * < \infty]$. $\mathbf{\Delta}$ is the symmetric difference operator for sets/languages. $L_1 =^a L_2 \Leftrightarrow$ card$(L_1 \mathbf{\Delta} L_2) \leq a$. ('$=^0$' denotes, then, ordinary set equality.) $L_1 \neq^a L_2$ means that it is *not* the case that $L_1 =^a L_2$.

'$\subset$' denotes 'is a *proper* subset of', and '$\supset$' denotes 'is a *proper* superset of'. Set theoretically, as in [Hal74], we treat sequences as functions, and, in general, functions, finite, partial or total, as *single-valued sets of ordered pairs.*[4] Hence, we can and do meaningfully compare them with '$\subseteq$', '$\subset$', '$\supseteq$', and '$\supset$'. It *follows*, for example, that, if $T$ if a text, '$\tau \subset T$' means that 'the finite sequence $\tau$ is an initial segment of the infinite sequence $T$'.

The quantifier '$\exists^\infty \tau$' means 'there exists infinitely many $\tau$'.

We use '|' to mean 'such that'.

**Definition 3** *Suppose* $\mathbf{F}$ *is a learning function and $T$ is a text. We say* $\mathbf{F}(T)$ converges *(written:* $\mathbf{F}(T)\Downarrow$*)* $\Leftrightarrow \{\mathbf{F}(\tau) \mid \tau \subset T\}$ *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,* $\mathbf{F}(T)$ *is undefined.*

[3]Decorations are subscripts, superscripts, primes, and the like.

[4]Hence, the sequence of numbers $w_0, w_1, w_2, \ldots$ is identified with function $f$ such that, for each $i \in \mathbf{N}$, $f(i) = w_i$, and this $f$ is also identified with its graph $\{(i, f(i)) \mid i \in \mathbf{N}\}$.

**Definition 4** *A language learning function,* $\mathbf{F}$*,* $\mathbf{TxtFex}^a_b$-identifies *an r.e. language* $L \Leftrightarrow$ $(\forall$ *texts* $T$ *for* $L)[\mathbf{F}(T)\Downarrow =$ *a set of cardinality* $\leq b$ *and* $(\forall p \in \mathbf{F}(T))[W_p =^a L]]$.

In $\mathbf{TxtFex}^a_b$-identification the $b$ is a "bound" on the number of final grammars and the $a$ a "bound" on the number of anomalies allowed in these final grammars. As above, a "bound" of $*$ just means *un*bounded, but *finite.*

**Definition 5** $\mathbf{TxtFex}^a_b$ *denotes the class of all* classes $\mathcal{L}$ *of languages such that some learning function* $\mathbf{TxtFex}^a_b$*-identifies each language in* $\mathcal{L}$*.*

Intuitively, $\mathcal{L} \in \mathbf{TxtFex}^a_b \Leftrightarrow$ there is an algorithm $\mathbf{p}$, computing a learning function $\mathbf{F}$, such that, if $\mathbf{p}$ is given any listing $T$ of any language $L \in \mathcal{L}$ , it outputs a sequence of grammars *converging* in a non-empty *set*, $\mathbf{F}(T)$, of no more than $b$ grammars, and each of these grammars makes no more than $a$ mistakes in generating $L$, i.e., if $\mathbf{p}$ is given any listing of any language $L \in \mathcal{L}$ , it outputs a sequence of grammars, and, past some point in this sequence, each grammar seen (over and over) is from a set of no more than $b$ grammars and each of these "final" grammars makes no more than $a$ mistakes in generating $L$.

$\mathbf{TxtFex}^0_1$-identification is equivalent to Gold's [Gol67] seminal notion of *identification*, also referred to as **TXTEX**-identification in [CL82] and (indirectly) as **INT** in [OW82b, OW82a, OSW86b].

$\mathbf{TxtFex}^a_1$-identification is just $\mathbf{TXTEX}^a$-identification from [CL82]. For $n > 0$, $\mathbf{TxtFex}^0_n$-identification is just our notion of $\mathbf{TXTFEX}_n$-identification from [Cas86]. Osherson and Weinstein [OW82a] were the first to define $\mathbf{TxtFex}^0_*$ and $\mathbf{TxtFex}^*_*$; they called them **BEXT** and **BFEXT**, respectively.

It is common in the literature use $\mathbf{TXTEX}^a_b$ to mean the special case of $\mathbf{TXTEX}^a$ where the total number of changes of output (or *mind changes*) is bounded above by $b$. *N.B. The* $b$ *in* $\mathbf{TxtFex}^a_b$ *has a totally different meaning from the* $b$ *in* $\mathbf{TXTEX}^a_b$*; the former is a bound on the number of different programs an associated machine eventually vacillates between in the limit; the latter is a bound on mind changes for convergence to a single final program.*

**Definition 6** *A text* $T$ *is* recursive $\Leftrightarrow$ $T$*, as a function:* $\mathbf{N} \rightarrow (\mathbf{N} \cup \{\#\})$*, is computable.*[5]

Learning power under $\mathbf{TxtFex}^a_b$-identification might be affected if one requires success only on all *recursive* texts for a language. This is interesting since, if, for example, the universe is *completely* algorithmic then all *real* language texts generated by parents for their children *are* recursive![6]

**Definition 7** *A language learning function,* $\mathbf{F}$*,* $\mathbf{RecTxtFex}^a_b$-identifies *an r.e. language* $L \Leftrightarrow$ $(\forall$ recursive *texts* $T$ *for* $L)[\mathbf{F}(T)\Downarrow =$ *a set of cardinality* $\leq b$ *and* $(\forall p \in \mathbf{F}(T))[W_p =^a L]]$.

**Definition 8** $\mathbf{RecTxtFex}^a_b$ *denotes the class of all* classes $\mathcal{L}$ *of languages such that some learning function* $\mathbf{RecTxtFex}^a_b$*-identifies each language in* $\mathcal{L}$ *.*

It is interesting to consider what happens to learning power if the final programs/grammars for $\mathbf{TxtFex}^a_b$-identification are required to be "nearly" minimal size, hence, even more likely to fit in ones head.

Let mingrammar$(L)$ denote $\min(\{p \mid W_p = L\})$.

---

[5]The r.e. languages are characterized as those which are the content of some recursive text [Rog67].

[6]See further discussion in Section 3 below.

**Definition 9** $\mathbf{F}$ $\mathbf{TxtMfex}^a_b$-identifies *a class of languages* $\mathcal{L}$ $\Leftrightarrow$ $(\exists$ *recursive* $h)(\forall L \in \mathcal{L})[\mathbf{F}\ \mathbf{TxtFex}^a_b\text{-identifies } L\ \wedge\ (\forall T \textit{ for } L)(\forall p \in \mathbf{F}(T))[p \leq h(\text{mingrammar}(L))]]$.

$h$ in Definition 9 plays the role of a computable amount by which the final programs can be larger than minimal size. This size restriction of course does not hold in general, and, for $\mathbf{TxtMfex}^0_1$-identification, it is not as severe as requiring that the final program be strictly minimal size. Mathematically $\mathbf{TxtMfex}^a_b$-identification is well-behaved, e.g., it turns out not to depend on the choice of acceptable system; it also does not depend on the choice of Blum program size measure [Blu67b] (by his recursive-relatedness result in [Blu67b]). The lack of dependence on the choice of acceptable system is in contrast with the variant of $\mathbf{TxtMfex}^0_1$-identification in which we require $h$ to be the identity function (see [CJS94b]). The study of learning nearly minimal size programs began with [Fre75] in the context of learning programs for functions (see also [Kin77, Che81, Che82, Fre90]).

**Definition 10** $\mathbf{TxtMfex}^a_b = \{\mathcal{L} \mid (\exists \mathbf{F})[\mathbf{F}\ \mathbf{TxtMfex}^a_b\textit{-identifies } \mathcal{L}]\}$.

Similarly we may define $\mathbf{RecTxtMfex}^a_b$*-identification* and $\mathbf{RecTxtMfex}^a_b$ as $\mathbf{TxtMfex}^a_b$-identification and $\mathbf{TxtMfex}^a_b$, respectively, restricted to recursive texts (see Definitions 7 and 8 above).

Next, for mathematical completeness and interest, are introduced the cases of success criteria for which the number of final grammars is possibly infinite, not necessarily finite as it is for $\mathbf{TxtFex}^a_b$-identification. Definitions 11 and 12 are from [CL82]. The $a \in \{0, *\}$ cases were independently introduced in [OW82a, OW82b].

The quantifier '$\forall^\infty k$' means 'for all but finitely many $k \in \mathbf{N}$'.

**Definition 11** $\mathbf{F}$ $\mathbf{TxtBc}^a$-identifies $L \Leftrightarrow (\forall \textit{ texts } T \textit{ for } L)(\forall^\infty k)[W_{\mathbf{F}(T[k])} =^a L]$.

**Definition 12** $\mathbf{TxtBc}^a$ *denotes the class of all classes* $\mathcal{L}$ *of languages such that some learning function* $\mathbf{TxtBc}^a$*-identifies each language in* $\mathcal{L}$.

$\emptyset$ denotes the empty set of natural numbers.

Fix *canonical indexings* of the *finite* sets of natural numbers and of the finite initial segments of texts each 1-1 onto $\mathbf{N}$ [Rog67, MY78].[7] In the following finite sets and segments are sometimes *identified* with their corresponding canonical indices. Hence, a reference to a *least* finite set or segment really refers to a finite set or segment with least canonical index. Also, *when we compare finite sets or segments by* $<, \leq, \ldots$ *we are comparing their corresponding canonical indices.*

$\langle \cdot, \cdot \rangle$ denotes a fixed *pairing function* [Rog67], a computable, surjective and injective mapping from $\mathbf{N} \times \mathbf{N}$ into $\mathbf{N}$.

For $A \subseteq \mathbf{N}$, $\overline{A}$ denotes $(\mathbf{N} - A)$, the *complement* of $A$.

We let $\mathbf{F}$ (with or without decorations) range over learning functions.

# 3 Results on Vacillation in Learning

This section presents our main results regarding the vacillatory learning criteria of the present paper.

In this section we defer proofs of three results until we have the benefit of some of the concepts and results from Sections 4 and 5 below. Section 6 further below contains the three deferred proofs.

[7] The canonical index of a finite set or segment is, then, a numerical code of it.

The definition of $\mathbf{TxtFex}_b^a$-identification (Definition 5 above in Section 2) requires success for *each* order of data presentation. For each non-empty *r.e.* language, there are continuum many such orders (texts) [Hal74], yet only countably many *recursive* ones (since there are only countably many Turing Machine programs for computing the recursive texts [Rog67]). In a completely computable universe (which ours might be), there are really only *recursive* texts available to be presented to learning machines. Of course the universe *may* be such that, while all the language learners are computable, there are some non-computable phenomena too. As noted in [OSW86b], since the utterances of children's caretakers depend heavily on external environmental events, such influences might introduce a random component into naturally occurring texts. It is, then, interesting and important to compare learning power where success is required on *all* texts with the cases where it is only required on all *recursive* texts.

Wiehagen [Wie77] essentially notes that $\mathbf{RecTxtFex}_1^0 = \mathbf{TxtFex}_1^0$ (a related result was first proved in [BB75]), and [CL82] essentially observes that $(\forall a)[\mathbf{RecTxtFex}_1^a = \mathbf{TxtFex}_1^a]$. We have, more generally,

**Corollary 1**

1. $(\forall a, b)[\mathbf{RecTxtFex}_b^a = \mathbf{TxtFex}_b^a]$.
2. $(\forall a, b)[\mathbf{RecTxtMfex}_b^a = \mathbf{TxtMfex}_b^a]$.

Hence, for all the vacillatory learning criteria of the present paper, it makes no difference in learning power whether or not we restrict the texts to be recursive! By contrast, for $\mathbf{TxtBc}^a$, learning procedures *can* exploit the assumption that they are receiving recursive texts; for $\mathbf{TxtBc}^a$, the restriction to recursive text *does* make a difference in learning power [CL82, Fre85].

The proof of Corollary 1 immediately above is deferred to Section 6 further below since it employs Theorems 4 and 7 from Section 5 below.

The topic of learning nearly minimal size programs/grammars is treated in greater depth in [CJS94b]. Herein we present results about such criteria only in the contexts of recursive text (Corollary 1 just above) and of restrictions on learning functions (Section 5 below). There is a small amount of additional discussion below in Section 7.

$\mathbf{RecTxtFex}_1^0 = \mathbf{TxtFex}_1^0$ entails that, if a given learning function $\mathbf{F}$ $\mathbf{RecTxtFex}_1^0$-identifies some class, *some* learning function $\mathbf{F}'$ will $\mathbf{TxtFex}_1^0$-identify it. However, by the following proposition, in some cases we can*not* have $\mathbf{F}' = \mathbf{F}$.

**Proposition 1** *There is a learning function which* $\mathbf{RecTxtFex}_1^0$*-identifies a language which it fails to* $\mathbf{TxtFex}_1^0$*-identify.*

Proof. Let $K = \{\, p \mid p \in W_p \,\}$, a well know r.e., not recursive set [Rog67]. Suppose $k$ is a recursive function with range $K$ [Rog67]. Let $K_s = \{\, k(s') \mid s' < s \,\}$. $C_A$ denotes the *characteristic function of* $A \subseteq \mathbf{N}$, the function 1 on $A$ and 0 off $A$. Clearly $(\forall x)[\lim_{s\to\infty} C_{K_s}(x) = C_K(x)]$. Let $S_\tau =$ the set of all $x \in$ the domain of $\tau$ such that $C_{K_{\|\tau\|}}$ agrees with $\tau$ on all inputs $\leq x$. Let

$$\text{agree}(\tau) = \begin{cases} 0, & \text{if } S_\tau = \emptyset; \\ 1 + \max(S_\tau), & \text{otherwise.} \end{cases}$$

It is easy to see that

$$[T = C_K \Rightarrow \lim_{\tau \subset T} \text{agree}(\tau) = \infty]$$

and that

$$[T \neq C_K \Rightarrow \lim_{\tau \subset T} \mathrm{agree}(\tau) < \infty].$$

Let $p_0$ be a program such that $W_{p_0} = \{0, 1\}$. Let

$$\tau^{-} = \begin{cases} \emptyset, & \text{if } \tau = \emptyset; \\ \tau', & \text{if } \tau = \tau' \diamond x. \end{cases} \tag{2}$$

Let

$$\mathbf{F}(\tau) = \begin{cases} \mathrm{agree}(\tau), & \text{if agree}(\tau) > \mathrm{agree}(\tau^{-}); \\ p_0, & \text{otherwise.} \end{cases}$$

Let $L = \{0, 1\}$. Clearly $C_K$ is a non-recursive text $T$ for $L$ such that $\mathbf{F}(T)\not\Downarrow$, yet $\mathbf{F}$ on any recursive text for $L$ converges to $p_0$, a program for $L$. □

Next is our main theorem. It says that, for each $n > 0$, some classes of languages can be algorithmically learned (in the limit) by converging to *up to* $n+1$ different, exactly correct grammars; *but* these classes can*not* be learned by converging (in the limit) to up to $n$ different grammars, where the up to $n$ grammars are each allowed to have a finite number of anomalies! Allowing one more grammar in the limit makes a big difference in learning power.

Hence, it is *possible* that, for some $n > 0$, people have evolved language learning strategies that exploit the greater learning power achieved by converging in the limit to up to $n+1$ rather than to up to $n$ grammars (see a critical discussion in Section 7 below).

**Theorem 1** *Suppose* $n > 0$. *Let* $\mathcal{L}_{n+1} =$

$$\{ L \mid L \textit{ is } \infty \ \wedge\ (\exists e_0, \ldots, e_n)[W_{e_0} = \ldots = W_{e_n} = L \ \wedge\ (\forall^{\infty} \langle x, y \rangle \in L)[y \in \{ e_0, \ldots, e_n \}]] \}.$$

*Then* $\mathcal{L}_{n+1} \in (\mathbf{TxtFex}^0_{n+1} - \mathbf{TxtFex}^*_n)$.

The detailed proof of Section 1 immediately above is deferred to Section 6 below since it depends, in part, on Definitions 16 and 19 and Theorem 5 in Section 5 below.

The reader may note that the languages in the class $\mathcal{L}_{n+1}$ from Theorem 1 just above have an intriguing self-referential character. It is useful to discuss this feature a bit in the interest of anticipating and answering a possible objection to the use of self-reference in witnessing the separation result of Theorem 1.

In the proof of Theorem 1, to handle the self-referential character of $\mathcal{L}_{n+1}$, we employ the $(n+1)$-ary recursion theorem, a folk theorem generalizing the Kleene Recursion Theorem [Rog67, Page 214] and the Smullyan Double Recursion Theorem [Smu61]; it is also a consequence of our Operator Recursion Theorem [Cas74], an infinitary analog of the finite-arity recursion theorems.

Intuitively, the $(n+1)$-ary recursion theorem provides a means for transforming any sequence of $n+1$ programs $p_0, \ldots, p_n$ into a corresponding sequence of programs $e(p_0), \ldots, e(p_n)$ such that each $e(p_i)$ first creates quiescent copies of $e(p_0), \ldots, e(p_n)$ (including a self copy, a copy of $e(p_i)$ itself), and, then, each $e(p_i)$ runs $p_i$ on the quiescent copies of $e(p_0), \ldots, e(p_n)$ any together with any externally given input. Each $e(p_i)$, in effect, has complete (low level) knowledge of $e(p_0), \ldots, e(p_n)$ (including self knowledge, knowledge of $e(p_i)$ itself), and $p_i$ represents how $e(p_i)$ *uses* its self knowledge, its knowledge of the other $e(p_j)$'s, and its knowledge of the external world. Infinite regress is not required since each $e(p_i)$ creates the copies of $e(p_0), \ldots, e(p_n)$ *externally* to itself. One mechanism to achieve this creation is a generalization of the self replication trick isomorphic to that employed by single-celled organisms [Cas74]. Another is for the programs $e(p_0), \ldots, e(p_n)$ to look in a common mirror to see which programs they are. [Cas94] provides a tutorial on thinking about and applying recursion theorems.[8] Herein, our application of the $(n+1)$-ary recursion theorem (to

[8]See [RC94] for discussion and applications of recursion theorems in severely resource-limited contexts.

prove Theorem 1) will be informal and the sequence $p_0, \ldots, p_n$ will be implicit.

Now for the possible objection: On the one hand, we argue above that Theorem 1 suggests a possibility regarding human language learning; on the other hand, we prove it by self/other reference, and it is common to regard self-referential examples as *un*natural. For example, Gödel proved his famous Incompleteness Theorem by a self-reference argument [Göd86, Men86], and his self-referential sentence providing an unprovable truth of, for example, *First Order Peano Arithmetic* (**FOPA**) is not natural — no number or combinatorial theorist would care whether it was true or false.

We answer this objection about self-referential proofs of existence theorems, with the following

**Informal Thesis 1** *If a self-referential example witnesses the existence of a phenomenon, there are* natural *examples witnessing same!*

For this informal thesis we present a brief plausibility argument and one piece of empirical evidence. Plausibility: self-reference arguments lay bare *an* underlying *simplest* reason for the theorems they prove [Rog67, Cas94]; if a theorem is true for such a simple reason, the "space" of reasons for its truth may be broad enough to admit natural examples. Empirical: Although Gödel proved his famous First Incompleteness Theorem by a self-reference argument, many years afterwards, Paris and Harrington [PH77] and later Friedman [Sim85, Sim87] found quite natural examples of combinatorial truths of first order arithmetic not provable in **FOPA**.[9] In fairness, regarding the above informal thesis, we note, for example, that the Blum Speed-Up Theorem [Blu67a] was originally proved by a self-reference argument[10], but natural witnesses to even exponential speed-up have not (yet) been found. However, even the self-reference proofs of this result are fairly complicated; hence, one might expect that natural examples are especially hard to find.

For some theoretical work instigated by Barzdin and dealing, in part, with eliminating dependence on self-referential examples, see Fulk's work on robust function learning in [Ful90b].

**Corollary 2** $(\forall a)[\mathbf{TxtFex}_1^a \subset \mathbf{TxtFex}_2^a \subset \ldots \subset \mathbf{TxtFex}_*^a]$.

**Corollary 3 (Osherson and Weinstein [OW82a])** $\mathbf{TxtFex}_1^0 \subset \mathbf{TxtFex}_*^0$.

We announced in [Cas86] that we could prove $\mathbf{TxtFex}_1^0 \subset \mathbf{TxtFex}_2^0$ by analyzing Osherson and Weinstein's proof of the immediately preceding corollary. Under our direction Karen Ehrlich generalized the combinatorics of this proof to get $\mathbf{TxtFex}_2^0 \subset \mathbf{TxtFex}_3^0$. The combinatorics for this approach to the general case are unpleasant. [OSW86b] contains a recursion theorem proof of the immediately preceding corollary based on the proof in [OW82a], but the same combinatorial difficulties occur in attempting to generalize this proof. We sought a combinatorially cleaner self-reference proof. A later conversation about this with Royer led to Royer and Kurtz supplying us with essentially the self-referential sets we use in Theorem 1 above. We believe their self-referential examples are somewhat simpler than those we had been working with. They also supplied some of the crucial combinatorics for the diagonal argument that goes with a special case.

It is interesting to note that, if one modifies the definition of $\mathbf{TxtFex}_n^a$-identification to require that the learning function must converge to *exactly* $n$ grammars, then the hierarchy of Corollary 2 above collapses.[11]

If one restricts ones attention to languages which are the (pairing function coded) graphs of *total functions*, then it is essentially shown (the $a = 0$ case in [BP73] and the $a > 0$ cases in [CS83]) that

[9] See [RC94] for an example from complexity theory.

[10] See also Young's version in [You73] and our Operator Recursion Theorem variant in [Smi94].

[11] Just output every $n$-th grammar.

the hierarchy again collapses. Hence, in the case of "scientific inference," i.e., the case of learning programs for computable functions, there is no power in vacillation.[12]

Therefore, Corollary 2 above is very sensitive to minor perturbations. We should mention, however, that there *are* some interesting effects on learning power for vacillatory function learning wrought by bounding *suitably sensitive* measures of the computational complexity of the learning functions themselves [CJS94a] and by the introduction of noisy input data [CJS96].

The next proposition provides a dual to Theorem 1 above. There are classes which can be learned with one program in the limit and with up to $m+1$ anomalies in that program which can*not* be learned with finitely many programs in the limit, but with each having no more than $m$ anomalies.

**Proposition 2** $(\mathbf{TxtFex}_1^{m+1} - \mathbf{TxtFex}_*^m) \neq \emptyset$.

PROOF. We identify total functions $f$ with $\{\, \langle x, f(x)\rangle \mid x \in \mathbf{N}\,\}$. Let $\mathcal{L} = \{\, \text{total } f \mid \varphi_{f(0)} =^{m+1} f \,\}$. Clearly $\mathcal{L} \in \mathbf{TxtFex}_1^{m+1}$. Also, $\mathcal{L} \in \mathbf{TxtFex}_*^m$ together with Theorems 2.6 and 2.9 from [CS83] yields a contradiction. $\square$

In [BC93] it is shown that $\{\, L \mid L =^{n+1} \mathbf{N} \,\}$ also witnesses the separation of Proposition 2.

Clearly from Theorem 1 and Proposition 2 we have our main

**Corollary 4** $\mathbf{TxtFex}_b^a \subseteq \mathbf{TxtFex}_d^c \Leftrightarrow [b \leq d$ *and* $a \leq c]$.

In the just above corollary (Corollary 4) we see that all *and only* the obvious inclusions hold. Hence, allowing more anomalies, final grammars or both enhances learning power, but anomalies and final grammars cannot in general completely substitute for one another. For example, $\mathbf{TxtFex}_2^0$ is incomparable to $\mathbf{TxtFex}_1^1$. That is, there are classes which can be learned with no mistakes and up to two final grammars which cannot be learned with up to one mistake and one final grammar, and there are other classes which can be learned with up to one mistake and one final grammar which cannot be learned with no mistakes and up to two final grammars.

**Corollary 5 (Case and Lynes [CL82])** $\mathbf{TxtFex}_1^0 \subset \mathbf{TxtFex}_1^1 \subset \ldots \subset \mathbf{TxtFex}_1^*$.

Osherson and Weinstein [OW82a] independently showed the case of $\mathbf{TxtFex}_1^0 \subset \mathbf{TxtFex}_1^*$ from the previous corollary.

**Corollary 6 (Osherson and Weinstein [OW82a])** $\mathbf{TxtFex}_*^0 \subset \mathbf{TxtFex}_*^*$.

Next are spelled out the connections between $\mathbf{TxtFex}_b^a$ and $\mathbf{TxtBc}^{a'}$. Of course allowing infinitely many grammars in the limit is not so realistic for modeling language learning, but, nonetheless, it is mathematically interesting to make the comparisons.

**Proposition 3** $\mathbf{TxtBc}^0 - \mathbf{TxtFex}_*^* \neq \emptyset$.

[12]For computable functions $f$, one can think of input $x$ as coding a scientific experiment and the output $f(x)$ as coding the corresponding experimental result. In this way results about learning programs for functions can be interpreted as results about finding predictive explanations for phenomena — as results about scientific induction. For more on this see [BB75, CS83, CJS92, BCJS94, CJNM94, LW94]. Re the names of the learning criteria studied in the present paper, originally [CS83] '**Ex**' stood for 'explanatory', '**Fex**' stood for 'finitely explanatory', and **Bc** for 'behaviorally correct'.

Proof. As in the proof of Proposition 2, we identify total functions $f$ with $\{ \langle x, f(x) \rangle \mid x \in \mathbf{N} \}$. Let $\mathcal{L} = \{ \text{total } f \mid (\forall^\infty k)[\varphi_{f(k)} = f] \}$. Clearly $\mathcal{L} \in \mathbf{TxtBc}^0$. Also, $\mathcal{L} \in \mathbf{TxtFex}_*^*$ together with Theorems 2.12 and 3.1 from [CS83] yields a contradiction. □

**Remark 1** *Proposition 3 still holds even if we restrict* $\mathbf{TxtBc}^0$*-identification to recursive texts.*

The next theorem says that, in passing from learning finitely many anomalous grammars in the limit to learning infinitely many, one can eliminate $\frac{1}{2}$ the anomalies, and that that's optimal! This contrasts with the function learning case [CS83] where, by a result of Steel, one can so eliminate *all* of finitely many anomalies. Intuitively, in the present context, since one is missing in the input data the negative information, i.e., since one is missing $\approx \frac{1}{2}$ the information, one can eliminate $\frac{1}{2}$ the anomalies only.

**Theorem 2** $\mathbf{TxtFex}_*^m \subseteq \mathbf{TxtBc}^{m'} \Leftrightarrow m \leq 2.m'$*; furthermore,* $\{ L \mid L =^{2m+1} \mathbf{N} \} \in (\mathbf{TxtFex}_1^{2m+1} - \mathbf{TxtBc}^m)$.

In the immediately above theorem (Theorem 2) we see that some excluded inclusions are, at first glance, unexpected. Its proof is deferred to Section 6 further below since it depends on Theorem 3 in Section 4 below.

Clearly, we have the following

**Corollary 7 ([CL82])** *The class of co-finite sets is in* $(\mathbf{TxtFex}_1^* - \bigcup_{m \in \mathbf{N}} \mathbf{TxtBc}^m)$.

We have not yet worked out all the relationships analogous to those in Theorem 2 and Corollary 7 for the cases $\mathbf{TxtBc}^a$-identification is restricted to recursive texts. As noted above in this section, the restriction to recursive texts *does* affect $\mathbf{TxtBc}^a$-identification [CL82, Fre85].

## 4 Topological Results

We next present several useful results which can be described as topological. The exact connections to topology (actually, to Baire Category Theory and Banach-Mazur Games [Jec78]) we will not pursue herein, but, on that subject, the interested reader can consult [OSW83, OSW86b].

**Definition 13** *Suppose* $\sigma \subseteq \tau \subset T$*,* $T$ *a text. Then*

$$\mathbf{F}[\sigma, \tau] = \{ p \mid (\exists \sigma' \supseteq \sigma \mid \sigma' \subseteq \tau)[p = \mathbf{F}(\sigma')] \}$$

*and*

$$\mathbf{F}[\sigma, T] = \{ p \mid (\exists \sigma' \supseteq \sigma \mid \sigma' \subset T)[p = \mathbf{F}(\sigma')] \}.$$

Suppose $\sigma$ is an finite initial segment of a text $T$. Picture $\mathbf{F}$ being fed $T$ one element at a time and imagine watching the successive corresponding output programs. Then, for example, from Definition 13 immediately above, $\mathbf{F}[\sigma, T]$ is the set of all these output programs one sees *from* the time $\mathbf{F}$ is fed all of $\sigma$.

**Definition 14** $\sigma$ in $L \Leftrightarrow \text{content}(\sigma) \subseteq L$.

Just below is a variant of a fundamental lemma from [OW82a] convenient for this paper. An original, not so general version of this lemma is from [BB75] (see also [OSW83, OSW86b]). Variations on its proof will appear in other proofs.

**Lemma 1** *Suppose $L \in \mathcal{E}$. Suppose, for each text $T$ for $L$, an arbitrary $\sigma_T \subset T$ is chosen. Then, for these choices, let*

$$P = \bigcup_{T \text{ for } L} \mathbf{F}[\sigma_T, T].$$

*It follows that*

$$(\forall \sigma \textit{ in } L)(\exists \tau \supseteq \sigma \mid \tau \textit{ in } L)(\forall \tau' \supseteq \tau \mid \tau' \textit{ in } L)[\mathbf{F}(\tau') \in P]. \tag{3}$$

Proof. Suppose the hypotheses. Suppose for contradiction the negation of (3). Hence,

$$(\exists \sigma \text{ in } L)(\forall \tau \supseteq \sigma \mid \tau \text{ in } L)(\exists \tau' \supseteq \tau \mid \tau' \text{ in } L)[\mathbf{F}(\tau') \notin P]. \tag{4}$$

Let $T$ be a fixed text for L. We recursively define another text $T'$ for $L$ as follows. Let $\tau_0 = \sigma$ and $\tau_0' = \tau_0 \diamond T(0)$. Suppose (recursively) that $\tau_n$ and $\tau_n' \supset \sigma$ are defined and in $L$. By (4) we may take $\tau_{n+1}$ to be the least $\supseteq \tau_n'$ such that $[\tau_{n+1}$ in $L \ \wedge\ \mathbf{F}(\tau_{n+1}) \notin P]$. Let $\tau_{n+1}' = \tau_{n+1} \diamond T(n+1)$. Let $T' = \bigcup_{n \in \mathbf{N}} \tau_n'$. Clearly, $T'$ is a text for $L$ and $T' = \bigcup_{n \in \mathbf{N}} \tau_n$ too, with $\tau_0 \subset \tau_1 \subset \tau_2 \subset \ldots$ . By the choice of $\tau_n$'s, for each $n \in \mathbf{N}$, $\mathbf{F}(\tau_{n+1}) \notin P$. Therefore, $\mathbf{F}[\sigma_{T'}, T'] \not\subseteq P$, a contradiction. □

The $I = \mathbf{Fex}_1^0$ case of the following Theorem is from Angluin's [Ang80]. She calls the finite sets $D$ featured *tell tales*. The theorem witnesses a severe constraint called the *subset principle* on learning from positive data. See [Ang80, Ber85] regarding the importance of the subset principle for circumventing *overgeneralization* in learning languages *from positive data*. See [KLHM93, Wex93] for discussion regarding the possible connection between this subset principle and a more traditionally linguistically oriented one in [MW87].

We let $2* \stackrel{\text{def}}{=} *$.

**Theorem 3** *Suppose $I \in \{\mathbf{Fex}_b^a, \mathbf{Bc}^a\}$. Suppose $\mathbf{F}$ $\mathbf{Txt}I$-identifies $L$. Then*

$$(\exists D \textit{ finite } \subseteq L)(\forall L' \subseteq L \mid D \subseteq L' \ \wedge\ L' \neq^{2a} L)[\mathbf{F} \textit{ does not } \mathbf{Txt}I\textit{-identify } L']. \tag{5}$$

It would be interesting to have a complete characterization from Theorem 3. Progress was made in [BCJ96] where it is shown that, for any *uniformly decidable class of recursive languages* $\mathcal{L}$, a learning function $\mathbf{F}$ witnesses that $\mathcal{L}$ is in $\mathbf{TxtBc}^a$ $\Leftrightarrow$ each $L \in \mathcal{L}$ satisfies (5) above (for $I = \mathbf{Bc}^a$).[13]

To prove Theorem 3 it is useful to have the following combinatorial lemma whose proof is omitted by reason of being straightforward.

**Lemma 2** *Suppose $[A =^a B \ \wedge\ B =^a C]$. Then $A =^{2a} C$.*

Proof of Theorem 3. Suppose the hypotheses. For each $T$ for $L$, choose a suitably large $\sigma_T \subset T$ that

$$(\forall \tau \supseteq \sigma_T \mid \tau \subset T)[W_{\mathbf{F}(\tau)} =^a L].$$

Let

$$P = \bigcup_{T \text{ for } L} F[\sigma_T, T].$$

[13] This complements a related characterization in [Ang80] of the uniformly decidable classes of recursive languages in $\mathbf{TxtFex}_1^0$. [BCJ96] provides a related characterization of the uniformly decidable classes of recursive languages in $\mathbf{TxtFex}_1^*$. [OSW86c, Page 30] characterizes learning by an agent which is *not necessarily algorithmic*. [Muk92, LZ92] contain characterizations of uniformly decidable classes of recursive languages for important special cases of $\mathbf{TxtFex}_1^0$. [dJK96] surprisingly characterizes the *r.e.* classes in $\mathbf{TxtFex}_1^0$, but by a condition more complicated than in the characterizations already mentioned.

Then $(\forall p \in P)[W_p =^a L]$. Hence, by Lemma 1,

$$(\exists \tau \supseteq \emptyset \mid \tau \text{ in } L)(\forall \tau' \supseteq \tau \mid \tau' \text{ in } L)[\mathbf{F}(\tau') \in P].^{14} \tag{6}$$

Let $D = \text{content}(\tau)$, a *finite* subset of $L$. Suppose $[D \subseteq L' \subseteq L \ \wedge \ L' \neq^{2a} L]$. Let $T'$ be a text for $L'$ such that $T' \supset \tau$. Then, since any $\tau' \subset T'$ is in $L$, we have by (6) that

$$(\forall \tau' \supseteq \tau \mid \tau' \subset T')[\mathbf{F}(\tau') \in P].$$

Hence, $(\forall \tau' \supseteq \tau \mid \tau' \subset T')[W_{\mathbf{F}(\tau')} =^a L \neq^{2a} L']$. Therefore, by Lemma 2, $(\forall \tau' \supseteq \tau \mid \tau' \subset T')[W_{\mathbf{F}(\tau')} \neq^a L']$. Hence, $\mathbf{F}$ does not $\mathbf{Txt}I$-identify $L'$. □ (Theorem 3)

Clearly in the proof of Theorem 3 there is *no* use of the computability of $\mathbf{F}$. The limitation Theorem 3 witnesses on learning from texts is purely topological having nothing to do with algorithmicity. Corollary 8 and the non-learnability half of Theorem 2 above, proved from Theorem 3, likewise do not depend on algorithmicity.

**Corollary 8 ([OW82a, CL82])** *Suppose $\mathcal{L}$ contains an infinite language $L$ and all its finite sublanguages. Then $\mathcal{L} \notin \mathbf{TxtBc}^*$. Hence, the class of regular languages $\notin \mathbf{TxtBc}^*$.*

Theorem 3 above does *not* imply that, if a learning function $\mathbf{TxtFex}_1^0$-identifies an infinite language, it must fail to $\mathbf{TxtFex}_1^0$-identify *each* proper sublanguage. In fact we have the following proposition a variant of which, regarding function learning, appears in [Cas94].

**Proposition 4** *There is in $\mathbf{TxtFex}_1^0$ an infinite r.e. collection of infinite languages of the form $\{ W_{e_0} \supset W_{e_1} \supset W_{e_2} \supset \dots \}$.*

Proof. By the Operator Recursion Theorem [Cas74], there is an infinite r.e. sequence of self-other referential programs $e_0, e_1, e_2, \dots$ such that, for each $i \in \mathbf{N}$,

$$W_{e_i} = \{ e_i, e_{i+1}, e_{i+2}, \dots \}.$$

We omit the straightforward verification. □

Gold [Gol67] proved Corollary 8 with $\mathbf{TxtFex}_1^0$ in place of $\mathbf{TxtBc}^*$ and was clearly concerned that his result meant that only rather puny language classes could be learned from positive data. However, Wiehagen [Wie77] presents a class of r.e. languages in $\mathbf{TxtFex}_1^0$ which contains a finite variant of each r.e. language. Wiehagen's class is obviously quite hefty. Angluin presents examples natural from the perspective of formal language theory that also are in $\mathbf{TxtFex}_1^0$ [Ang80, Ang82]. All these classes in $\mathbf{TxtFex}_1^0$ (of course) satisfy the subset principle (of Theorem 3), and, in particular, they are not closed under finite sublanguages as is the class of regular languages.

Suppose $\mathcal{N}$ is a class of natural languages learnable from text and which contains some language $L$ and also an infinitely different natural sublanguage $L'$ of $L$. For example, $L'$ might be the class of imperative sentences of $L$. Theorem 3 above causes no apparent problem since a finite tell tale $D$ for $L$ need not (and should not) be contained in $L'$. It may be useful for linguists to try to find such tell tales $D$'s for natural languages $L$. Of course such a $D$ shouldn't be contained in, for example, $L'$, the set of imperative sentences of $L$, but should, nonetheless, be salient empirically to the learning of $L$.

The following stability property is useful for studying the criteria $\mathbf{RecTxtFex}_b^a$.

[14] $\tau$ is, then, what is suggestively called a *locking sequence* [OSW86b].

**Definition 15** *Suppose* $L$ *is r.e. Then:* $L$ recursively $b$-stabilizes $\mathbf{F} \Leftrightarrow$

$$(\forall \textit{ recursive } T \textit{ for } L)(\exists D \mid \mathrm{card}(D) \leq b)[\mathbf{F}(T)\Downarrow = D].$$

The following lemma, useful to this paper, combines the topological with the algorithmic. It generalizes predecessors from [BB75, Ful85, OSW86b].

**Lemma 3** *Suppose* $L$ *is r.e. and recursively* $b$*-stabilizes* $\mathbf{F}$*. Then*

$$(\forall \sigma \textit{ in } L)(\exists D \mid \mathrm{card}(D) \leq b)(\exists \tau \supseteq \sigma \mid \tau \textit{ in } L)(\forall \tau' \supseteq \tau \mid \tau' \textit{ in } L)[\mathbf{F}(\tau') \in D]. \tag{7}$$

Proof. Suppose the hypothesis on $L$. Suppose for contradiction the negation of (7). Hence,

$$(\exists \sigma \text{ in } L)(\forall D \mid \mathrm{card}(D) \leq b)(\forall \tau \supseteq \sigma \mid \tau \text{ in } L)(\exists \tau' \supseteq \tau \mid \tau' \text{ in } L)[\mathbf{F}(\tau') \notin D]. \tag{8}$$

Let $T$ be a fixed recursive text for L. We recursively define another recursive text $T'$ for $L$ as follows. Let $\tau_0 = \sigma$ and $\tau_0' = \tau_0 \diamond T(0)$. Let $\tau_0''$ = the *shortest* $\subseteq \tau_0'$ such that $\mathrm{card}(\mathbf{F}[\tau_0'', \tau_0']) \leq b$. (For $b = *$, $\tau_n''$ will $= \emptyset$, for all $n \in \mathbf{N}$.) Let $D^0 = \mathbf{F}[\tau_0'', \tau_0']$. Suppose (recursively) that $\tau_n$, $\tau_n'$, and $\tau_n''$ are defined and in $L$, $\tau_n, \tau_n' \supset \sigma$, $\tau_n'' \subseteq \tau_n'$, and that $D^n = \mathbf{F}[\tau_n'', \tau_n']$. By (8) we may algorithmically find a $\tau_{n+1} \supseteq \tau_n'$ such that $[\tau_{n+1} \text{ in } L \ \wedge \ \mathbf{F}(\tau_{n+1}) \notin D^n]$. Let $\tau_{n+1}' = \tau_{n+1} \diamond T(n+1)$. Let $\tau_{n+1}''$ = the *shortest* $\subseteq \tau_{n+1}'$ such that $[\tau_{n+1}'' \supseteq \tau_n'' \ \wedge \ \mathrm{card}(\mathbf{F}[\tau_{n+1}'', \tau_{n+1}']) \leq b]$. Let $D^{n+1} = \mathbf{F}[\tau_{n+1}'', \tau_{n+1}']$. Let $T' = \bigcup_{n \in \mathbf{N}} \tau_n'$. Clearly, $T'$ is a recursive text for $L$ and $T' = \bigcup_{n \in \mathbf{N}} \tau_n$ too, with $\tau_0 \subset \tau_1 \subset \tau_2 \subset \ldots$ . By the choice of $\tau_n$'s, for each $n \in \mathbf{N}$, $\mathbf{F}(\tau_{n+1}) \notin D^n$. Therefore, $\mathbf{F}(T')\not\Downarrow$ to a set of cardinality $\leq b$, a contradiction to the hypothesis on $L$. □

# 5 Insensitive or Restricted Learning Functions

It is interesting to ask whether or not child language learning exhibits sensitivity to *the order or the timing of presentation of data.* We consider herein some mathematical versions of this question. Several mathematical definitions have been given for various different notions of *in*sensitivity to order, essentially for the case of $\mathbf{TxtFex}_1^0$-identification [BB75, WC80, SR84, Ful85, OSW86b, Ful90a].

We extend these definitions of insensitive or restricted learning functions naturally to the context of the vacillatory learning criteria of the present paper,[15] and we investigate the interesting mathematical questions of whether learning functions with these *in*sensitivities or restrictions thereby lose learning power. Answering many of these questions for the vacillatory criteria is much more difficult than for the $\mathbf{TxtFex}_1^0$ case.[16]

As noted above, we also apply some of our results in this section to help us prove results in this and other papers.

**Definition 16 (Wexler [WC80, OSW86b])** $\mathbf{F}$ *is called* set-driven $\Leftrightarrow$ $(\forall \sigma, \tau \mid \mathrm{content}(\sigma) = \mathrm{content}(\tau))[\mathbf{F}(\sigma) = \mathbf{F}(\tau)]$.

[15] For the so-called order independence notions (Definition 19 below), in the interest of conceptual parsimony, but without loss of generality in theorems, we render them purely syntactically rather as a mixture of syntactical and semantical (as their precursor notions are in the prior literature). The precursor notions required the final programs/grammars also to be correct, a *semantic* constraint which we eliminate from the definitions.

[16] It is in many cases especially difficult to prove that the *simultaneous* presence of several insensitivities leads to *no* loss of learning power. We had several painful experiences, for example, with subtly incorrect, alternative constructions to the one in the proof of Theorem 4 below.

[WC80] essentially notes that *set driven* learning functions are insensitive to *time* (unlike text learnability). The next restriction defined, in effect, provides some degree of sensitivity to timing.

**Definition 17 (Schäfer [SR84, OSW86b], Fulk [Ful85, Ful90a])** $\mathbf{F}$ *is called* partly set-driven *(synonym [Ful85, Ful90a]:* rearrangement independent*)* $\Leftrightarrow (\forall \sigma, \tau \mid \|\sigma\| = \|\tau\| \ \wedge \ \text{content}(\sigma) = \text{content}(\tau))[\mathbf{F}(\sigma) = \mathbf{F}(\tau)]$.

Intuitively, $\mathbf{F}$ is set-driven (respectively, partly set-driven) iff, for each $\sigma$, $\mathbf{F}(\sigma)$ depends only on the *content* of $\sigma$ (respectively, depends only on the *length and content* of $\sigma$).

Schäfer [SR84, OSW86b], first, and Fulk [Ful85, Ful90a], later, independently showed that set-driven learning functions can't $\mathbf{TxtFex}_1^0$-identify some classes of languages that unrestricted learning functions can, but that partly set-driven learning functions do not restrict learning power with respect to $\mathbf{TxtFex}_1^0$-identification. Fulk additionally showed that set-driven learning functions can't even $\mathbf{TxtBc}^0$-identify some languages classes in $\mathbf{TxtFex}_1^0$. He interprets the difference in power between set-driven and partly set-driven learning functions as witnessing the need for time $>$ the size of the content of the input to "think" about the input.

Osherson, Stob, and Weinstein [OSW86b] observed that the power of $\mathbf{TxtFex}_1^0$-identification on *infinite* r.e. languages is not limited by set-driveness.

The following definition presents a convenient term paralleling that from Definition 15 above.

**Definition 18** *A text* $T$ stabilizes $\mathbf{F} \Leftrightarrow \mathbf{F}(T){\Downarrow}$.

While identification of a language $L$ requires identification for *each* order of presentation of (text for) $L$, the *final* (correct) grammar(s) converged to may be different for different texts. As is, in effect, noted in [OSW86b], this would seem to be source of strength, since, for a learning machine to force the final grammars to be the same for each text, might involve its (algorithmically) recognizing *grammar equivalence*, i.e., recognizing $\{ \langle x, y \rangle \mid W_x = W_y \}$, but, as is well known [Rog67], this set is *not* algorithmically recognizable (r.e.) (nor is its complement).[17]

*Order independent* machines are insensitive to which text is used for $L$ in that their final grammars depend only on $L$, not on the order of presentation. Their grammars along the way can, of course, depend on the text.

If, for some $n > 0$ and for some $a$, humans $\mathbf{TxtFex}_{n+1}^a$-identify a language $L$ but do not $\mathbf{TxtFex}_n^a$-identify it, it is interesting, whether, nonetheless, *some* environments and corresponding texts for $L$ cause them to output fewer final conjectures than $n+1$. There is a corresponding and ostensibly weaker notion of order independence in which, for each text, the set of final grammars converged to is always *contained in* (but not necessarily equal to) some *finite* set of final grammars.

These order independence notions clearly capture a very different kind of insensitivity to order of data presentation than the set-driven notions above. The formal definition for our order independence notions immediately follows.

**Definition 19**

1. *We call a learning function,* $\mathbf{F}$*,* $b$-ary order independent $\Leftrightarrow$ $(\forall\ L$ *r.e.* $\mid$ some *text for* $L$ *stabilizes* $\mathbf{F})(\exists D$ *of cardinality* $\leq b)(\forall$ *texts* $T$ *for* $L)[\mathbf{F}(T){\Downarrow} = D]$.

2. *We call a learning function,* $\mathbf{F}$*,* weakly $b$-ary order independent $\Leftrightarrow$ $(\forall\ L$ *r.e.* $\mid$ some *text for* $L$ *stabilizes* $\mathbf{F})(\exists D$ *of cardinality* $\leq b)(\forall$ *texts* $T$ *for* $L)[\mathbf{F}(T){\Downarrow} \subseteq D]$.

[17] In fact, more importantly, since this set is $\Pi_2^0$-complete [Rog67], it is not even algorithmically recognizable by a limiting [Soa87] or mind-changing procedure (but its complement is).

Osherson, Stob, and Weinstein [OSW86b], adapting a related result of L. and M. Blum [BB75], essentially show that order independent learning functions can $\mathbf{TxtFex}_1^0$-identify the same classes of languages that unrestricted learning functions can.

The first theorem of the present section (Theorem 4 below) implies that learning power (with respect to $\mathbf{TxtFex}_b^a$-identification) is not decreased by restricting learning functions to be *simultaneously* partly set-driven and weakly $b$-ary order independent. Furthermore, it implies that one can also simultaneously circumvent the restriction to recursive texts. It generalizes parts of Fulk's Kitchen Sink Theorem [Ful90a, Theorem 13, Page 6] and [Ful85, Chapter 5, Theorem 21] which covered the $\mathbf{TxtFex}_1^0$ case only; however, the lift to Theorem 4 ostensibly requires a much more difficult proof.[18]

For non-trivially vacillatory criteria, it is open whether (full) order independence can be combined with partly set-driven without loss of learning power.

Theorem 4's proof is the most difficult of the present paper. Fortunately, the other proofs of theorems in this section are modifications, and/or simplifications of the proof of Theorem 4.

**Theorem 4** *There is an algorithm for transforming any $b$ and (algorithm for) a learning function $\mathbf{F}$ into a corresponding (algorithm for) a learning function $\mathbf{F}'$ such that*

1. *$\mathbf{F}'$ is both partly set-driven and weakly $b$-ary order independent and*

2. $(\forall$ *r.e.* $L)[\mathbf{F}$ $\mathbf{RecTxtFex}_b^a$*-identifies* $L \Rightarrow \mathbf{F}'$ $\mathbf{TxtFex}_b^a$*-identifies* $L]$.

Proof. Suppose pad is a 1-1 computable function such that $(\forall n, p)[W_{\mathrm{pad}(p,n)} = W_p]$ [MY78, Roy87]. Intuitively, $\mathrm{pad}(p,0), \mathrm{pad}(p,1), \mathrm{pad}(p,2), \ldots$ are just padded variants of program $p$ which have the same recognizing behavior as $p$ but which differ from one another syntactically.

Suppose $\mathbf{F}$ and $b$ are given. Define $\mathbf{F}'$ on $\tau$ thus. Set $n = \|\tau\|$ and $A = \mathrm{content}(\tau)$.

($*$ In the definition of $\mathbf{F}'(\tau)$ the only dependence on $\tau$ will be on $n$ and $A$ to make sure $\mathbf{F}'$ is partly set-driven. $*$)
Search for the *least* $\langle D^1, \sigma^1\rangle$ such that[19]

1. $\mathrm{card}(D^1) \leq b$,

2. $\mathrm{content}(\sigma^1) \subseteq A$, and

3. $(\forall \sigma' \supseteq \sigma^1 \mid \sigma' \leq n \ \wedge\ \mathrm{content}(\sigma') \subseteq A)[\mathbf{F}[\sigma^1, \sigma'] \subseteq D^1]$.[20]

($*$ Clearly such a $\langle D^1, \sigma^1\rangle$ will always exist since $\sigma^1$ may be chosen big enough not to be contained in any $\sigma' \leq n$. $*$)

($*$ Suppose $\tau$ in $L$. Clause 3 just above provides a bounded (by $n$) approximation to

$$(\forall \sigma' \supseteq \sigma^1 \mid \mathrm{content}(\sigma') \subseteq L)[\mathbf{F}[\sigma^1, \sigma'] \subseteq D^1]. \tag{9}$$

(9) is a useful stability condition. $*$)
Once $\langle D^1, \sigma^1\rangle$ is found:

[18] In the present paper we do not consider the restriction to so-called *prudence* [OSW86b], a primary concern of [Ful90a]. *Prudent learning functions* are those which never conjecture a grammar $p$ without being able to learn $W_p$. On that subject the interested reader may also wish to consult [JS95, KR88].

[19] It is useful to recall here that, from Section 2 above, $\langle \cdot, \cdot \rangle$ is a numerical pairing function *and* that we identify finite sets and initial segments of texts with their corresponding canonical indices (numbers). The word *least*, then refers to least *numerical* value.

[20] Again, it is useful to recall that, from Section 2 above, we identify finite initial segments of texts with their corresponding canonical indices (numbers). Hence, in the inequality, '$\sigma' \leq n$', we are treating $\sigma$ as its numerical canonical index.

set $i = 1$;
**while** [card$(D^i) > 1 \ \wedge$ a *least* $\langle D', \sigma' \rangle \leq n$ is found such that $D' \subset D^1 \ \wedge \ \sigma' \supset \sigma^i \ \wedge \ \text{content}(\sigma') \subseteq A \ \wedge \ (\forall \sigma'' \supseteq \sigma' \mid \sigma'' \leq n \ \wedge \ \text{content}(\sigma'') \subseteq A)[\mathbf{F}[\sigma', \sigma''] \subseteq D']]$[21]
**do** (∗ Pump down from $D^i$ and ratchet up from $\sigma^i$, preserving apparent stability. ∗)
increment $i$ by 1;
set $\langle D^i, \sigma^i \rangle = \langle D', \sigma' \rangle$
**endwhile**;
set $\mathbf{F}'(\tau) = \text{pad}(\mathbf{F}(\sigma^i), \langle D^1, \sigma^1 \rangle)$.

Clearly, by construction, $\mathbf{F}'$ is partly set-driven.

**Intuitive Discussion**:
Something like the while loop in (the algorithm for) $\mathbf{F}'$ is essential. It is crucial to establishing Claim 3 below. If stopping with a search for $\langle D^1, \sigma^1 \rangle$ sufficed, Theorem 1 above could not hold. Nothing like this while loop is needed to handle the cases of $\mathbf{TxtFex}_1^a$. The use of pad is a variant of its use in [Ful85, Ful90a] and serves below in the proof of $\mathbf{F}'$'s weak $b$-ary order independence in a combinatorially similar, subtle role.

Here's an intuitive way to think about this construction. Imagine a chimpanzee given an infinite collection of different kinds of sticks some of which can be joined together to make longer sticks. Each stick points overhead in a particular direction with respect to the vertical. Above the chimp, but out of its sight, is a bunch of bananas it would like to knock down with a suitably large joined together stick pointing in just the right direction to hit the bananas. We suppose that it can't tell when it has actually reached the bunch of bananas even thought it does (so the poor thing never knows when its succeeded and it never actually gets to eat the bananas). All it can tell is that some time after any choice of a (leaning) tower of sticks is not pointing quite right, one of the sticks will explode, knocking down all of the sticks above it, and it has to try again. The exploding sticks are quite like the injuries in a recursion-theoretic priority argument [Soa87].

The sticks correspond to the $\sigma$'s and their extensions, and one should think of them as segments of branches in an infinite branching, upward pointing tree similar to the finite branching (rightward pointing) one in [Rog67, Page 157]. For each input $\tau$ to $\mathbf{F}'$, when the while loop finishes, it provides some sequence of successively longer joined together sticks $\sigma^1 \subset \ldots \subset \sigma^m$, with $m$ the final value of $i$. A larger input to $\mathbf{F}'$, $\tau' \supset \tau$, may result in a different sequence of sticks, $\sigma_1 \subset \ldots \subset \sigma_{m'}$, from the while loop. The stick that exploded is in the $\sigma^i$ with least $i$ such that $\sigma^i \neq \sigma_i$. "Success" for the chimpanzee is described by Claim 1 just below. We continue this discussion after the statment of that claim.

**Claim 1** *If $L$ recursively $b$-stabilizes $\mathbf{F}$, then, for each text $T$ for $L$, there is a* maximum $j \geq 1$ *such that the algorithm for $\mathbf{F}'$ above on $T$* eventually *has stable values for $\langle D^1, \sigma^1 \rangle, \ldots, \langle D^j, \sigma^j \rangle$, i.e., values that are the same for (the algorithm for) $\mathbf{F}'$'s calculation of $\mathbf{F}'(\tau)$, for all but finitely many $\tau \subset T$. This $j$ will also be $\leq b$. Furthermore, if there is such a maximum $j$ for* some *text for $L$, values of this maximum $j$ and associated stable values of $\langle D^1, \sigma^1 \rangle, \ldots, \langle D^j, \sigma^j \rangle$ will be* independent *of the choice of text for $L$.*

[21] N.B. It is useful to recall here that, from Section 2 above, '$\subset$' denotes 'is a *proper* subset of', and '$\supset$' denotes 'is a *proper* superset of'.

**Continued Discussion**:
If $L$ recursively $b$-stabilizes $\mathbf{F}$ and $T$ is a text for $L$, then this claim does *not* imply that, for all but finitely many $\tau \subset T$, the while loop on input $\tau$ stops with the same $\langle D^1, \sigma^1 \rangle, \ldots, \langle D^j, \sigma^j \rangle$ — only that the while loop stops with $\langle D^1, \sigma^1 \rangle, \ldots, \langle D^i, \sigma^i \rangle$, for some $i \geq j$. Success for the chimpanzee discussed above is the stabilizing on $\sigma^1 \subset \ldots \subset \sigma^j$, but even after this stability is reached, any sticks $\sigma^i$, for $i > j$, returned by the while loop will have an "explosion" on some longer input to $\mathbf{F}'$.

Suppose $L$ recursively $b$-stabilizes $\mathbf{F}$ and $T$ is a text for $L$. Suppose $j$ is as in the just previous paragraph. As $\mathbf{F}$ is being fed successively longer initial segments of $T$, eventually $\sigma^j$ is reached. We also like to think about the changing of $\sigma^i$'s subsequently found, where $i > j$, as a *flickering flame* above $\sigma^j$. Stability implies that, for infinitely many $\tau \subset T$, the flame *may* die down to exactly the level of $\sigma^j$ itself; however, for all but finitely many $\tau \subset T$, it does not dip below or destroy $\sigma^j$.

Proof of Claim 1. Suppose $L$ recursively $b$-stabilizes $\mathbf{F}$. Then by Lemma 3,

$$(\exists D \mid \mathrm{card}(D) \leq b)(\exists \sigma \supseteq \emptyset \mid \sigma \text{ in } L)(\forall \sigma' \supseteq \sigma \mid \sigma' \text{ in } L)[\mathbf{F}(\sigma') \in D]. \tag{10}$$

(The algorithm for) $\mathbf{F}'$ on texts for $L$ will eventually stabilize in its choice of $\langle D^1, \sigma^1 \rangle$ to be the *same for each* $T'$ *for* $L$: it will stabilize its choice of $\langle D^1, \sigma^1 \rangle$ to be the least $\langle D, \sigma \rangle$ satisfying (10). This is since, for all but finitely many $\tau \subset T$, $\|\tau\|$ and content$(\tau)$ will be big enough to find counterexamples to all the *finitely* many $\langle D', \sigma' \rangle <$ this least $\langle D, \sigma \rangle$ satisfying (10). Of course, once a $\langle D', \sigma' \rangle$ is rejected for being $\langle D^1, \sigma^1 \rangle$, it's not picked up again by (the algorithm for) $\mathbf{F}'$ on bigger input since counterexamples don't go away for bigger input. Once the choice of $\langle D^1, \sigma^1 \rangle$ has stabilized on a $T$ for $L$, say, on all sufficiently large $\tau \subset T$; on such suitably large $\tau$, the while loop eventually terminates with a final value for $i$, say $i_\tau$, which is $\leq b$ since card$(D^1) \leq b$, and the while loop looks for *proper* subsets of the $D^i$s. Clearly, as above, on suitably large $\tau \subset T$, there is a maximum $i \leq$ the while loop's $i_\tau$'s with $\langle D^1, \sigma^1 \rangle, \ldots, \langle D^i, \sigma^i \rangle$ eventually stable, and, also clearly, this maximum $i$ is independent of texts for $L$. □ (Claim 1)

**Claim 2** $\mathbf{F}'$ *is weakly b-ary order independent.*

Proof of Claim 2. Suppose $T$ for $L$ stabilizes $\mathbf{F}'$. We need to show, then, that ($\exists D$ of cardinality $\leq b$)$[\bigcup_{T' \text{ for } L} \mathbf{F}'(T')\Downarrow \subseteq D]$. Once (the algorithm for) $\mathbf{F}'$ on (successively longer $\tau \subset$) $T$ rejects a candidate for $\langle D^1, \sigma^1 \rangle$, it cannot rechoose that candidate later since counterexamples to the stability demanded of $\langle D^1, \sigma^1 \rangle$ do not go away. $\mathbf{F}'$ on $T$ outputs programs of the form pad$(\mathbf{F}(\sigma^i), \langle D^1, \sigma^1 \rangle)$, where $\langle D^1, \sigma^1 \rangle$ is a candidate for stability at the first level, so to speak. Since, by assumption just above, $T$ *does* stabilize $\mathbf{F}'$, for some *finite* $D$, $\mathbf{F}'(T)\Downarrow =$ D, and, then, since pad is 1-1, the $\langle D^1, \sigma^1 \rangle$ argument to it cannot take on infinitely many values as $\mathbf{F}'$ is fed T. Since $\mathbf{F}'$ can't jump back to rejected previous choices of $\langle D^1, \sigma^1 \rangle$, (the algorithm for) $\mathbf{F}'$ on $T$ eventually finds a stable value for $\langle D^1, \sigma^1 \rangle$. Hence, by a simple restatement of the proof of Claim 1 above, there is a maximum $i$ such that (the algorithm for) $\mathbf{F}'$ on $T$ eventually has stable values for $\langle D^1, \sigma^1 \rangle, \ldots, \langle D^i, \sigma^i \rangle$, and the value of $i$ is independent of texts for $L$. Let imax denote this maximum $i$. Hence, $(\forall \tau \supseteq \sigma^{\mathrm{imax}} \mid \tau \text{ in } L)[\mathbf{F}(\tau) \in D^{\mathrm{imax}}]$. Therefore,

$$\bigcup_{T' \text{ for } L} \mathbf{F}'(T')\Downarrow \subseteq \mathrm{pad}(D^{\mathrm{imax}}, \langle D^1, \sigma^1 \rangle). \tag{11}$$

This latter set of programs has cardinality $\leq b$ since $D^{\text{imax}}$ does. Therefore, $\mathbf{F}'$ is weakly $b$-ary order independent. □ (CLAIM 2)

N.B. There is no guarantee that (11) (in the immediately above proof) is an equality since we may have that, on some $T$ for $L$, for all but finitely many $\tau \subset T$, for the corresponding $\sigma^{i_\tau}$'s from the while loop, the programs $\text{pad}(\mathbf{F}(\sigma^{i_\tau}), \langle D^1, \sigma^1 \rangle)$ miss some values in $\text{pad}(D^{\text{imax}}, \langle D^1, \sigma^1 \rangle)$.

**Claim 3** *Suppose $L$ recursively $b$-stabilizes $\mathbf{F}$. Let* imax *be the maximum $i$ from Claim 1 (independent of the choice of text for $L$). Let*

$$D^{\text{Rec}} = \bigcup_{\substack{\textit{recursive } T \textit{ for } L \\ T \supset \sigma^{\text{imax}}}} \mathbf{F}(T).$$

*Equivalently,*

$$D^{\text{Rec}} = \{\, \mathbf{F}(\tau) \mid \tau \textit{ in } L \ \wedge \ \tau \supseteq \sigma^{\text{imax}} \ \wedge \ (\exists \textit{ recursive } T \textit{ for } L)(\exists^{\infty} \tau' \subset T)[\mathbf{F}(\tau') = \mathbf{F}(\tau)] \,\}. \quad (12)$$

*Then $D^{\text{Rec}} = D^{\text{imax}}$.*

PROOF OF CLAIM 3. Suppose the Hypotheses. Clearly $D^{\text{Rec}} \subseteq D^{\text{imax}}$. It remains to show $D^{\text{imax}} \subseteq D^{\text{Rec}}$. In that interest, suppose $p \in D^{\text{imax}}$. We will show $p \in D^{\text{Rec}}$. By the maximality of imax,

$$\neg(\exists \sigma' \supseteq \sigma^{\text{imax}} \mid \sigma' \text{ in } L)(\forall \sigma'' \supseteq \sigma' \mid \sigma'' \text{ in } L)[\mathbf{F}[\sigma', \sigma''] \subseteq D^{\text{imax}} - \{\, p \,\}].$$

Hence,

$$(\forall \sigma' \supseteq \sigma^{\text{imax}} \mid \sigma' \text{ in } L)(\exists \sigma'' \supseteq \sigma' \mid \sigma'' \text{ in } L)[\mathbf{F}(\sigma'') = p]. \quad (13)$$

Let $T$ be a fixed recursive text for L. We recursively define another recursive text $T'$ for $L$ as follows. Let $\tau_0 = \sigma^{\text{imax}}$ and $\tau_0' = \tau_0 \diamond T(0)$. Suppose (recursively) that $\tau_n$ and $\tau_n' \supset \sigma^{\text{imax}}$ are defined and in $L$. By (13) we may algorithmically find a $\tau_{n+1} \supseteq \tau_n'$ such that $[\tau_{n+1} \text{ in } L \ \wedge \ \mathbf{F}(\tau_{n+1}) = p]$. Let $\tau_{n+1}' = \tau_{n+1} \diamond T(n+1)$. Let $T' = \bigcup_{n \in \mathbf{N}} \tau_n'$. Clearly, $T'$ is a recursive text for $L$ and $T' = \bigcup_{n \in \mathbf{N}} \tau_n$ too, with $\tau_0 \subset \tau_1 \subset \tau_2 \subset \ldots$ . By the choice of $\tau_n$'s, for each $n \in \mathbf{N}$, $\mathbf{F}(\tau_{n+1}) = p$. Hence, $T'$ is a recursive text for $L$ such that $[T' \supset \sigma^{\text{imax}} \ \wedge \ \mathbf{F}$ on $T'$ outputs $p$ infinitely often$]$. Therefore, by (12), $p \in D^{\text{Rec}}$. □ (CLAIM 3)

**Claim 4** $(\forall$ *r.e.* $L)[\mathbf{F}\ \mathbf{RecTxtFex}_b^a$*-identifies* $L \Rightarrow \mathbf{F}'\ \mathbf{TxtFex}_b^a$*-identifies* $L]$.

PROOF OF CLAIM 4. Suppose $L$ is r.e. and $\mathbf{F}$ $\mathbf{RecTxtFex}_b^a$-identifies $L$. It remains to show $\mathbf{F}'$ $\mathbf{TxtFex}_b^a$-identifies $L$. Clearly, $L$ recursively $b$-stabilizes $\mathbf{F}$. Therefore, by Claim 1, a maximum imax exists with eventually stable values for $\langle D^1, \sigma^1 \rangle, \ldots, \langle D^{\text{imax}}, \sigma^{\text{imax}} \rangle$ independent of texts for $L$ in the operation of (the algorithm for) $\mathbf{F}'$. Clearly, $(\forall p \in D^{\text{Rec}})[W_p =^a L]$. By Claim 3, $D^{\text{Rec}} = D^{\text{imax}}$, so we have $(\forall p \in D^{\text{imax}})[W_p =^a L]$. Hence, $(\forall p \in \text{pad}(D^{\text{imax}}, \langle D^1, \sigma^1 \rangle))[W_p =^a L]$. Therefore, by (11), $\mathbf{F}'$ $\mathbf{TxtFex}_b^a$-identifies $L$. □ (CLAIM 4)

□ (THEOREM 4)

The next theorem (Theorem 5) implies that learning power for *infinite* r.e. languages (with respect to $\mathbf{TxtFex}_b^a$-identification) is not decreased by restricting learning functions to be *simultaneously* (completely) set-driven and weakly $b$-ary order independent. Furthermore, it implies that one can also simultaneously circumvent the restriction to recursive texts.

**Theorem 5** *There is an algorithm for transforming any $b$ and (algorithm for) a learning function $\mathbf{F}$ into a corresponding (algorithm for) a learning function $\mathbf{F}'$ such that*

1. *$\mathbf{F}'$ is both set-driven and weakly $b$-ary order independent and*

2. *$(\forall \infty$ r.e. $L)[\mathbf{F}$ $\mathbf{RecTxtFex}_b^a$-identifies $L \Rightarrow \mathbf{F}'$ $\mathbf{TxtFex}_b^a$-identifies $L]$.*

Proof. Modify (the algorithm for) $\mathbf{F}'$ in the proof above of Theorem 4 by setting $n = \mathrm{card}(\mathrm{content}(\tau))$ (instead of setting $n = \|\tau\|$). Since for *infinite* $L$, this $n$ grows, one can apply the rest of the proof of Theorem 4 *mutatis mutandis*[22]. □

Royer and Kurtz suggested to us that the use of set-driven learning functions could simplify the proof of at least a special case of Theorem 1 in Section 3 below, and Jun Tarui pointed out to us that weak $b$-ary order independence would further simplify proving Theorem 1. The proof herein of Theorem 1 makes use of Theorem 5.

We believe it is not possible to replace weak $b$-ary order independence by $b$-ary order independence in Theorems 4 and 5 above, contrary to our slightly overzealous claims in [Cas88]. However, we have the following result (Theorem 6) with Fulk (who is not responsible for the possibly incorrect claims in [Cas88]). Theorem 6 implies that learning power (with respect to $\mathbf{TxtFex}_b^a$-identification) is not decreased by simultaneously restricting learning functions to be (fully) $b$-ary order independent and circumventing the restriction to recursive texts. It also implies one can also simultaneously have a technical property we call *determination by single text* (part 2 of the Theorem).

This theorem sees application in [CJS94b], and the (full) $b$-ary order independence is important for that application.

**Theorem 6 (Case and Fulk)** *There is an algorithm for transforming any $b$ and (algorithm for) a learning function $\mathbf{F}$ into a corresponding (algorithm for) a learning function $\mathbf{F}'$ such that*

1. *$\mathbf{F}'$ is $b$-ary order independent,*

2. *$(\forall$ r.e. $L)[\mathbf{F}'$ $\mathbf{TxtFex}_b^a$-identifies $L$ on* some *text for $L \Rightarrow \mathbf{F}'$ $\mathbf{TxtFex}_b^a$-identifies $L]$, and*

3. *$(\forall$ r.e. $L)[\mathbf{F}$ $\mathbf{RecTxtFex}_b^a$-identifies $L \Rightarrow \mathbf{F}'$ $\mathbf{TxtFex}_b^a$-identifies $L]$.*

Proof. Suppose $\mathbf{F}$ and $b$ are given. The algorithm for $\mathbf{F}'$ is much like that in the proof of Theorem 4 above with some exceptions as noted below. $\tau^-$ is as defined in (2) above. In defining $\mathbf{F}'$ on $\tau$, we assume we have iteratively (on successively larger $\tau' \subset \tau$) kept a priority queue of programs/grammars, which queue is initially empty. Proceed initially as in the algorithm in the proof of Theorem 4 above, but, if the value of $\langle D^1, \sigma^1 \rangle$ associated with $\tau$ is $\neq$ the value of $\langle D^1, \sigma^1 \rangle$ associated with $\tau^-$, empty the priority queue, and output $\|\tau\|$; else: continue down through the end of the while loop and then let

$$\sigma = \sigma^{i_\tau}, \tag{14}$$

where, as in the proof of Theorem 4, $i_\tau$ is the final value of $i$ from the while loop for input $\tau$; next in increasing order of $\sigma'$ such that $\sigma'$ in $A$, $\sigma' \leq n$, and $\sigma' \supseteq \sigma$, ($\sigma$ from (14)), put $\mathbf{F}(\sigma')$ on the *tail* of the priority queue; when that is all done, output the *front* of the priority queue.

The outputting of $\|\tau\|$ upon witnessing an instability in the choice of $\langle D^1, \sigma^1 \rangle$ is essentially a combinatorial device from [BB75], and it plays the role pad did in the proof of Theorem 4 above, similarly controlling thrashing in the choice of $\langle D^1, \sigma^1 \rangle$ when some $T$ for $L$ stabilizes $\mathbf{F}'$. This

[22] With appropriate (and straightforward) changes being made.

makes $\mathbf{F}'$ weakly $b$-ary order independent. Clearly, if some text for $L$ stabilizes $\mathbf{F}'$, by the priority queue mechanism, for any $T$ for $L$, $\mathbf{F}'(T)\Downarrow = D^{\mathrm{imax}}$, where $D^{\mathrm{imax}}$ is from the proof of Theorem 4. Hence, $\mathbf{F}'$ is $b$-ary order independent. Clause 2 of Theorem 6 clearly follows. To prove clause 3 of Theorem 6, one can apply appropriate portions of the proof of Theorem 4 *mutatis mutandis.* □

We do not know if there are analogs of Theorems 4 and 5 above for $\mathbf{TxtMfex}_b^a$-identification. The use of pad in the proofs of those theorems wreaks havoc with program/grammar size. However, we do have the next three Theorems, the first of which sees application in [CJS94b].

These theorems say we can have, for $\mathbf{TxtMfex}_b^a$-identification, without loss of learning power, either

1. $b$-ary order independence, determination by single text, and circumvention of the restriction to recursive texts (Theorem 7);
2. partly set-driven learning functions and circumvention of the restriction to recursive texts (Theorem 8); *or*
3. (completely) set-driven learning functions and circumvention of the restriction to recursive texts (Theorem 9), *but this latter conjunction is without loss of learning power for infinite languages only.*

**Theorem 7 (Case and Jain)** *There is an algorithm for transforming any $b$ and (algorithm for) a learning function $\mathbf{F}$ into a corresponding (algorithm for) a learning function $\mathbf{F}'$ such that*

1. $\mathbf{F}'$ *is $b$-ary order independent,*
2. $(\forall$ *r.e.* $L)[\mathbf{F}'$ $\mathbf{TxtMfex}_b^a$*-identifies $L$ on* some *text for $L \Rightarrow \mathbf{F}'$ $\mathbf{TxtMfex}_b^a$-identifies $L$], and*
3. $(\forall$ *r.e.* $L)[\mathbf{F}$ $\mathbf{RecTxtMfex}_b^a$*-identifies $L \Rightarrow \mathbf{F}'$ $\mathbf{TxtMfex}_b^a$-identifies $L$].*

Proof. The proof of Theorem 6 just above suffices *mutatis mutandis.* □

The next theorem was independently noticed by Jain.

**Theorem 8 (Case, Jain)** *There is an algorithm for transforming any $b$ and (algorithm for) a learning function $\mathbf{F}$ into a corresponding (algorithm for) a learning function $\mathbf{F}'$ such that*

1. $\mathbf{F}'$ *is partly set-driven and*
2. $(\forall$ *r.e.* $L)[\mathbf{F}$ $\mathbf{RecTxtMfex}_b^a$*-identifies $L \Rightarrow \mathbf{F}'$ $\mathbf{TxtMfex}_b^a$-identifies $L$].*

Proof. The proof of Theorem 4 above with the elimination of any mention of pad and weak $b$-ary order independence, *mutatis mutandis*, suffices to prove the present theorem. □

Similarly, the proof of Theorem 5 above may be modified along the lines suggested in the proof of Theorem 8 just above to prove

**Theorem 9** *There is an algorithm for transforming any $b$ and (algorithm for) a learning function $\mathbf{F}$ into a corresponding (algorithm for) a learning function $\mathbf{F}'$ such that*

1. $\mathbf{F}'$ *is set-driven and*
2. $(\forall \infty$ *r.e.* $L)[\mathbf{F}$ $\mathbf{RecTxtMfex}_b^a$*-identifies $L \Rightarrow \mathbf{F}'$ $\mathbf{TxtMfex}_b^a$-identifies $L$].*

We expect that the theorems of this section will be generally useful for work in the area.

# 6 Proofs Deferred from Section 3

In Section 3 above we deferred proofs of three results until we had the benefit of some of the concepts and/or results from Sections 4 and 5 which follow Section 3. The present section contains those deferred proofs, and for convenience, we restate each result being proved.

Clearly, the second conclusion of Theorem 4 and the third conclusion of Theorem 7, each in Section 5 above, yield the following

**Corollary 1**

1. $(\forall a, b)[\mathbf{RecTxtFex}_b^a = \mathbf{TxtFex}_b^a]$.
2. $(\forall a, b)[\mathbf{RecTxtMfex}_b^a = \mathbf{TxtMfex}_b^a]$.

As we noted in Section 3 above, the proof of the next theorem depends, in part, on Definitions 16 and 19 and Theorem 5 from Section 5 above.

**Theorem 1** Let $\mathcal{L}_{n+1} =$

$$\{ L \mid L \text{ is } \infty \ \wedge \ (\exists e_0, \ldots, e_n)[W_{e_0} = \ldots = W_{e_n} = L \ \wedge \ (\forall^\infty \langle x, y \rangle \in L)[y \in \{ e_0, \ldots, e_n \}]] \}.$$

Then $\mathcal{L}_{n+1} \in (\mathbf{TxtFex}_{n+1}^0 - \mathbf{TxtFex}_n^*)$.

Proof. Clearly $\mathcal{L}_{n+1} \in \mathbf{TxtFex}_{n+1}^0$.[23]

Suppose for contradiction that $\mathbf{F}$ $\mathbf{TxtFex}_n^*$-identifies $\mathcal{L}_{n+1}$. Each member of $\mathcal{L}_{n+1}$ is infinite; hence, thanks to Theorem 5 in Section 5 above, we may suppose without loss of generality that $\mathbf{F}$ is set-driven *and* weakly $n$-ary order independent. Therefore, in particular, we may write $\mathbf{F}(D)$ for $\mathbf{F}(\sigma)$, where $D = \text{content}(\sigma)$. By implicit application of a padded version of the $n+1$-ary recursion theorem there are *distinct* self-other referential $e_0, e_1, \ldots, e_n$ defining $W_{e_0}, W_{e_1}, \ldots, W_{e_n}$, respectively, in successive stages $s$ as follows.[24]

For each $i \leq n$: let $W_{e_i,s} =$ the finitely much of $W_{e_i}$ defined *before* stage $s$ described below; also set $W_{e_i,0} = \emptyset$. Go to stage 0.

> **begin** *stage s*
>   **if** $\text{card}(\{ \mathbf{F}(W_{e_0,s}), \mathbf{F}(W_{e_1,s}), \ldots, \mathbf{F}(W_{e_n,s}) \}) \leq n$
>     **then**
>       for each $i \leq n$, set $W_{e_i,s+1} = W_{e_i,s} \cup \{ \langle s, e_i \rangle \}$
>     **else**
>       for each $i \leq n$, set $W_{e_i,s+1} = [[\bigcup_{j \leq n} W_{e_j,s}] \cup \{ \langle s, e_0 \rangle \}]$
>   **endif**;
>   go to stage $s+1$
> **end** (* *stage s* *).

[23] The role of self-reference in this proof is, in part, to make this positive portion of the theorem immediate while scarcely affecting the difficulty of the negative portion below.

[24] The padding is just to make $e_0, e_1, \ldots, e_n$ syntactically pairwise distinct. It should be clear in the staging construction how each $e_i$ significantly uses its knowledge of $e_0, \ldots, e_i, \ldots, e_n$.

*Case (1).* $(\forall^\infty s)[\mathrm{card}(\{\,\mathbf{F}(W_{e_0,s}), \mathbf{F}(W_{e_1,s}), \ldots, \mathbf{F}(W_{e_n,s})\,\}) \leq n]$. Then each of $W_{e_0}, W_{e_1}, \ldots, W_{e_n} \in \mathcal{L}_{n+1}$, yet they are pairwise $\neq^*$. Hence, since this is Case (1), at each sufficiently large stage $s$, for at least one of the $(n+1)$ $i$'s $\leq n$, program/grammar $\mathbf{F}(W_{e_i,s})$ fails to generate a finite variant of $W_{e_i}$. Therefore, for at least one $i \leq n$, $(\exists^\infty s)[W_{\mathbf{F}(W_{e_i,s})} \neq^* W_{e_i}]$. Hence, *this* $W_{e_i}$ is *not* $\mathbf{TxtFex}^*_n$-identified by $\mathbf{F}$, a contradiction.

*Case (2).* $(\exists^\infty s)[\mathrm{card}(\{\,\mathbf{F}(W_{e_0,s}), \mathbf{F}(W_{e_1,s}), \ldots, \mathbf{F}(W_{e_n,s})\,\}) = n+1]$ (say at stages $s_0 < s_1 < s_2 < \ldots$ ). Then $W_{e_0} = W_{e_1} = \ldots = W_{e_n} \in \mathcal{L}_{n+1}$. Let $\vec{W_{e_i,s}}$ be an increasing order enumeration of $W_{e_i,s}$. Hence, $\vec{W_{e_i,s}}$ is also a finite initial segment of a text. Let $T_i = \vec{W_{e_i,s_0}} \diamond \vec{W_{e_i,s_1}} \diamond \vec{W_{e_i,s_2}} \diamond \ldots$ . Clearly $T_i$ is a text for $W_{e_i}$, which equals $W_{e_0}$. Since $\mathbf{F}$ is weakly $n$-ary order independent, there is a set $D$ of cardinality $\leq n$ such that

$$\bigcup_{i \leq n} \mathbf{F}(T_i)\Downarrow \subseteq D. \tag{15}$$

However, since this is Case (2) and by the choice of $s_0, s_1, s_2, \ldots$ , the left hand side of (15) has cardinality $> n$, a contradiction. □

As we noted in Section 3 above, the proof of the next theorem depends on Theorem 3 in Section 4 above.

**Theorem 2** $\mathbf{TxtFex}^m_* \subseteq \mathbf{TxtBc}^{m'} \Leftrightarrow m \leq 2.m'$; furthermore, $\{\,L \mid L =^{2m+1} \mathbf{N}\,\} \in (\mathbf{TxtFex}^{2m+1}_1 - \mathbf{TxtBc}^m)$.

PROOF. This proof employs previously unpublished techniques used to prove a similar result for $\mathbf{TxtFex}^a_1$ in [CL82].

Suppose $\mathbf{F}$ $\mathbf{TxtFex}^{2m}_*$-identifies $\mathcal{L}$. We will construct an $\mathbf{F}'$ which $\mathbf{TxtBc}^m$-identifies $\mathcal{L}$, and, then, it will suffice to prove the furthermore clause.

Define $\mathbf{F}'$ on $\tau$ thus. First calculate $p = \mathbf{F}(\tau)$. By Kleene's S-m-n Theorem [Rog67], find $p_\tau$ such that $W_{p_\tau} = ((W_p \cup \mathrm{content}(\tau)) -$ the $m$ least numbers not in $\mathrm{content}(\tau))$. Output $p_\tau$.

Suppose $T$ is a text for $L \in \mathcal{L}$. Let $D = \mathbf{F}(T)\Downarrow$. Hence, $(\forall p \in D)[W_p =^{2m} L]$. For all sufficiently large $\tau \in T$, $p = \mathbf{F}(\tau) \in D$ and $p_\tau$ patches any mistakes of omission of $p$; furthermore, $p_\tau$ removes $m$ elements, including up to $m$ of the mistakes of commission of $p$ (if any) and perhaps in the process creating new mistakes of omission.

*Case (1).* The number of mistakes of commission in such a $p$ is $\geq m$. Of course this number of mistakes is $\leq 2m$. Then $p_\tau$ removes $m$ of these mistakes of commission leaving a residue of $\leq m$ errors.

*Case (2).* The number $m'$ of mistakes of commission in such a $p$ is $< m$. Then $p_\tau$ removes *all* these errors of commission, but creates $m - m'$ new errors (of omission); however, this number is still $\leq m$.

In each case, for such $p$, $p_\tau$ has $\leq m$ errors. Therefore, $\mathbf{F}'$ $\mathbf{TxtBc}^m$-identifies $L \in \mathcal{L}$.

Let $\mathcal{L} = \{\,L \mid L =^{2m+1} \mathbf{N}\,\}$. Clearly, $\mathcal{L} \in \mathbf{TxtFex}^{2m+1}_1$. Suppose for contradiction $\mathcal{L} \in \mathbf{TxtBc}^m$ as witnessed by learning function $\mathbf{F}$. Hence, in particular, $\mathbf{F}$ $\mathbf{TxtBc}^m$-identifies $\mathbf{N}$. Therefore, by Theorem 3 in Section 4 above,

$$(\exists D \text{ finite })(\forall L \mid D \subseteq L \ \wedge\ L \neq^{2m} \mathbf{N})[\mathbf{F} \text{ does not } \mathbf{TxtBc}^m\text{-identify } L]. \tag{16}$$

Pick $L \supseteq D$ such that $\mathrm{card}(\overline{L}) = 2m+1$. Then $L \in \mathcal{L}$, but, by (16), $\mathbf{F}$ does not $\mathbf{TxtBc}^m$-identify $L$, a contradiction.

□

# 7 Concluding Remarks

In this section we discuss briefly computable universe hypotheses; present some critical discussion about the applicability to human language learning of Gold-style models and our main theorem (Theorem 1 in Section 3 above); and sketch some areas for future investigation.

We have considered (among other possibilities) computable models of learning on computable data sequences. The whole universe or humanly significant portions of it may be computable and/or discrete. Such possibilities are taken seriously, for example, in [Zus69, Tof77, TM87, Fey82, Cas92b, Cas86, CRS94]. In a discrete, *random* universe with only computable probability distributions for its behavior (e.g., a discrete, quantum mechanical universe), the *expected* behavior will still be computable [dMSS56, Gil72, Gil77].[25] In such a universe any beings (e.g., human) who do cognition, including language learning and scientific induction, will be subject to the constraint that at least their expected behavior will be computable; hence, any theorems about computable learning agents will inform, to some extent, about the possible behaviors of those beings. It would appear that human genetic programs make use of error correction in an attempt to circumvent "random" influences including from the quantum mechanical level. It is plausible that human cognitive programs built on top of the wetware the genetic programs partly construct do likewise. Hence, computability of cognition may be a pretty good model.

Even if cognition is computable (although, perhaps too complicated for mere humans to figure out how its done), there are still problems realistically modeling human language learning with Gold's paradigm. [MB72, MB73] present empirical evidence that semantics in addition to positive information may be essential to human language learning. It seems clear that denotation and social reinforcers play crucial roles in the human case — but not in Gold's paradigm. In [Cas86] the report on Chapter 6 of [Ful85] is partly motivated by treating negative information as a more mathematically tractable possible substitute for semantic information. [McN66] notes that homes in which parents do supply improvements to child utterances (a subtle form of correction or negative information), there is increased *speed* of language acquisition. It is not clear if the relation is causal, but Theorem 22 in [BCJ95] implies there are cases where a significant improvement in language learning *speed* (as calibrated by number of mind-changes required to reach a single final correct grammar) results from the presence of minimal negative information. Largely unexplored, but of some interest, is the extension of [BCJ95] to $\mathbf{TxtFex}_b^a$-identification.

We originally suggested [Cas88] on the basis of our main corollary (Corollary 4 to Theorem 1, each in Section 3 above) that Gold's model be extended to embrace the success criteria $\mathbf{TxtFex}_b^a$ for "small" values of $a$ and $b$. We consider next a possible difficulty. In the proof of Theorem 1, for each $\mathbf{F}$, the associated set(s) $W_{e_0}, W_{e_1}, \ldots, W_{e_n}$ may, in some cases, differ considerably in computational complexity from one another, and Osherson pointed out to us that there is no apparent corresponding vacillation in human language performance. However, in the proof of Theorem 1, for each $\mathbf{F}$, the associated set(s) $W_{e_0}, W_{e_1}, \ldots, W_{e_n}$ are each actually recursive; hence, for each $\mathbf{F}$, there is a Blum Complexity Measure $\Phi$ [Blu67a, HU79] such that $\Phi_{e_0} = \Phi_{e_1} = \ldots = \Phi_{e_n}$; therefore, if performance were measured by such a $\Phi$, vacillatory learning would increase learning power but without a corresponding vacillation in performance. Technical questions remain open regarding which stronger quantificational variants of the argument in the just previous sentence can be made. In another direction, we note that the proof of Theorem 1 permits a modification so that the relative density of output of all the final programs/grammars but one is as small as we like. Hence, the

[25] Sources such as [Pen89, Pen94] sadly seem to have overlooked the important result in [dMSS56] that the expected I/O behavior of a Turing machine with random oracle subject to a computable probability distribution is computable (and constructively so).

performance vacillation may exist, but significant degradations in articulateness potential might be confined to rare episodes. Even if such episodes do not exist for people, they might be tolerated in an artificial system.

In spite of the limitations to date of modeling human language learning with (extensions of) Gold's paradigm, we believe that many of the theorems (e.g., Theorem 3 in Section 4 above) in this area do, nonetheless, give some insights. The state of the art is weakly analogous to modeling the thermodynamics of fluids without taking into account van der Waal's forces: one may still get some understanding of the reality so modeled.

Speaking of Theorem 3: It would be mathematically interesting to explore what happens to the subset principle for **TxtBc**-identification restricted to *recursive* texts.

It is interesting to place further feasibility restrictions on the criteria of success. As noted in Section 5 above, [CJS94b] studies $\mathbf{TxtMfex}_b^a$-identification, the restricted variant of $\mathbf{TxtFex}_b^a$-identification which requires that final programs/grammars be nearly minimal size. For language learning, bounding complexity of learning machines as in [DS86] or [CJS94a] largely remains to be explored. Translating relative solvability results into relative feasibility results, as in [WZ92], would be very interesting to pursue in the context of the present paper. In Section 5 there are several results about no loss of learning power in passing from some learning function $\mathbf{F}$ to an insensitive or restricted learning function $\mathbf{F}'$. How does the complexity of such $\mathbf{F}'$'s compare to that of $\mathbf{F}$? If the complexity of $\mathbf{F}'$, in some cases, must be significantly greater than that of $\mathbf{F}$, then one could plausibly conjecture that child language learning is importantly sensitive to order of data presentation.

Can we get versions of our separation results robust in the sense of [Ful90b]?

Much of the work in Gold-style learning theory on success criteria extending Gold's is motivated by attempts to assuage the negative results in this area. [Kir92] mentions a common argument to the effect that very strong negative results about language learnability in [Gol67] provide evidence that human language learning must involve some innately stored information! The negative results suggest, among other things,

1. that general purpose learning is not possible and

2. that alleged human general purpose learning is an illusion brought about by our having innate information stored for a large and varied collection of domains [GBC$^+$91, Spe94].

   In the practical context of robot planning, Drew McDermott [McD92] says, "Learning makes the most sense when it is thought of as filling in the details in an algorithm that is already nearly right." In the context of function learning, [CKKK95] provides several models of learning from examples *together with approximately correct programs*. Included are models in which the maximal probability of learning *all* the computable functions is proportional to how tightly the approximately correct programs envelope the data. Unexplored, but very interesting, is how to provide such models for language learning from positive data. Success might provide some insight into the form of innate knowledge for human language learning.

## References

[Ang80] D. Angluin. Inductive inference of formal languages from positive data. *Information and Control*, 45:117–135, 1980.

[Ang82] D. Angluin. Inference of reversible languages. *Journal of the ACM*, 29:741–765, 1982.

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

[BC93] G. Baliga and J. Case. Learnability: Admissible, co-finite, and hypersimple sets. In *Proceedings of the 20th International Colloquium on Automata, Languages and Programming*, volume 700 of *Lecture Notes in Computer Science*, pages 289–300. Springer-Verlag, Berlin, Lund, Sweden, July 1993. Journal version in press for *Journal of Computer and System Sciences*, 1996.

[BCJ95] G. Baliga, J. Case, and S. Jain. Language learning with some negative information. *Journal of Computer and System Sciences*, 51:273–285, 1995.

[BCJ96] G. Baliga, J. Case, and S. Jain. Synthesizing enumeration techniques for language learning. Technical Report eC-TR-96-003, Electronic Archive for Computational Learning Theory (http://ecolt.informatik.uni-dortmund.de/), 1996.

[BCJS94] G. Baliga, J. Case, S. Jain, and M. Suraj. Machine learning of higher order programs. *Journal of Symbolic Logic*, 59(2):486–500, 1994.

[Ber85] R. Berwick. *The Acquisition of Syntactic Knowledge*. MIT Press, Cambridge, MA, 1985.

[BH70] R. Brown and C. Hanlon. Derivational complexity and the order of acquisition in child speech. In J. R. Hayes, editor, *Cognition and the Development of Language*. Wiley, 1970.

[Blu67a] M. Blum. A machine independent theory of the complexity of recursive functions. *Journal of the ACM*, 14:322–336, 1967.

[Blu67b] M. Blum. On the size of machines. *Information and Control*, 11:257–265, 1967.

[BP73] J. Barzdin and K. Podnieks. The theory of inductive inference. In *Proceedings of the Mathematical Foundations for Computer Science*, pages 9–15, 1973.

[Bra71] M. Braine. On two types of models of the internalization of grammars. In D. Slobin, editor, *The Ontogenesis of Grammar: A Theoretical Symposium*. Academic Press, NY, 1971.

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

[Cas86] J. Case. Learning machines. In W. Demopoulos and A. Marras, editors, *Language Learning and Concept Acquisition*. Ablex Publishing Company, 1986.

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

[Cas92a] J. Case. The power of vacillation in language learning. Technical Report 93-08, University of Delaware, 1992.

[Cas92b] J. Case. Turing machine. In Stuart Shapiro, editor, *Encyclopedia of Artificial Intelligence*. John Wiley and Sons, New York, NY, second edition, 1992.

[Cas94] J. Case. Infinitary self-reference in learning theory. *Journal of Experimental and Theoretical Artificial Intelligence*, 6:3–16, 1994.

[Che81] K. Chen. *Tradeoffs in Machine Inductive Inference*. PhD thesis, Computer Science Department, SUNY at Buffalo, 1981.

[Che82] K. Chen. Tradeoffs in the inductive inference of nearly minimal size programs. *Information and Control*, 52:68–86, 1982.

[CJNM94] J. Case, S. Jain, and S. Ngo Manguelle. Refinements of inductive inference by Popperian and reliable machines. *Kybernetika*, 30:23–52, 1994.

[CJS92] J. Case, S. Jain, and A. Sharma. On learning limiting programs. *International Journal of Foundations of Computer Science*, 3(1):93–115, 1992.

[CJS94a] J. Case, S. Jain, and A. Sharma. Complexity issues for vacillatory function identification. *Information and Computation*, 1994. To appear.

[CJS94b] J. Case, S. Jain, and A. Sharma. Vacillatory learning of nearly minimal size grammars. *Journal of Computer and System Sciences*, 49(2):189–207, October 1994.

[CJS96] J. Case, S. Jain, and F. Stephan. Vacillatory and BC learning on noisy data. Technical Report eC-TR-96-002, Electronic Archive for Computational Learning Theory (http://ecolt.informatik.uni-dortmund.de/), 1996.

[CKKK95] J. Case, S. Kaufmann, E. Kinber, and M. Kummer. Learning recursive functions from approximations. In *Proceedings of the Second European Conference on Computational Learning Theory*, volume 904 of *Lecture Notes in Artificial Intelligence*, pages 140–153, March 1995. Journal version to appear in *Journal of Computer and System Sciences* (Special Issue EuroCOLT'95).

[CL82] J. Case and C. Lynes. Machine inductive inference and language identification. In *Proceedings of the 9–th Annual Colloquium on Automata, Languages, and Programming*, Lecture Notes in Computer Science, 140, pages 107–115. Springer–Verlag, Berlin, July 1982.

[CRS94] J. Case, D. Rajan, and A. Shende. Representing the spatial/kinematic domain and lattice computers. *Journal of Experimental and Theoretical Artificial Intelligence*, 6:17–40, 1994.

[CS78] J. Case and C. Smith. Anomaly hierarchies of mechanized inductive inference. In *Proceedings of the 10–th Annual Symposium on the Theory of Computing*, pages 314–319, July 1978.

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

[dJK96] D. de Jongh and M. Kanazawa. Angluin's thoerem for indexed families of r.e. sets and applications. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory, Desenzano del Garda, Italy*, page 193, July 1996.

[dMSS56] K. deLeeuw, E. Moore, C. Shannon, and N. Shapiro. Computability by probabilistic machines. *Automata Studies, Annals of Math. Studies*, 34:183–212, 1956.

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

[Fey82] R. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6/7), 1982.

[Fre75] R. Freivalds. Minimal Gödel numbers and their identification in the limit. Lecture Notes in Computer Science, 32, pages 219–225. Springer-Verlag, Berlin, 1975.

[Fre85] R. Freivalds. Recursiveness of the enumerating functions increases the inferrability of recursively enumerable sets. *Bulletin of the European Association for Theoretical Computer Science*, 27:35–40, 1985.

[Fre90] R. Freivalds. Inductive inference of minimal programs. In M. Fulk and J. Case, editors, *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 3–20. Morgan Kaufmann Publishers, Inc., August 1990.

[Ful85] M. Fulk. *A Study of Inductive Inference machines.* PhD thesis, SUNY at Buffalo, 1985.

[Ful90a] M. Fulk. Prudence and other conditions on formal language learning. *Information and Computation*, 85:1–11, 1990.

[Ful90b] M. Fulk. Robust separations in inductive inference. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science*, pages 405–410, St. Louis, Missouri 1990.

[GBC+91] C. Gallistel, A. Brown, S. Carey, R. Gelman, and F. Keil. Lessons from animal learning for the study of cognitive development. In S. Carey and R. Gelman, editors, *Epigenesis of Mind: Essays on Biology and Cognition*, pages 3–37. Erlbaum, Hillsdale, 1991.

[Gil72] J. Gill. *Probabilistic Turing Machines and Complexity of Computation.* PhD thesis, University of California, Berkeley, 1972.

[Gil77] J. Gill. Computational complexity of probabilistic Turing machines. *SIAM Journal on Computing*, 6:675–695, 1977.

[Gle86] L. Gleitman. Biological dispositions to learn language. In W. Demopoulos and A. Marras, editors, *Language Learning and Concept Acquisition.* Ablex Publ. Co., 1986.

[Göd86] K. Gödel. On formally undecidable propositions of Principia Mathematica and related systems I. In S. Feferman, editor, *Kurt Gödel. Collected Works. Vol. I*, pages 145–195. Oxford Univ. Press, 1986.

[Gol67] E. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[Hal74] P. Halmos. *Naive Set Theory.* Springer-Verlag, NY, 1974.

[HU79] J. Hopcroft and J. Ullman. *Introduction to Automata Theory Languages and Computation.* Addison-Wesley Publishing Company, 1979.

[Jec78] Thomas Jech. *Set Theory.* Academic Press, NY, 1978.

[JL88] P. Johnson-Laird. *The Computer and the Mind: An Introduction to Cognitive Science.* Harvard Univesity Press, Cambridge, MA, 1988.

[JS95] S. Jain and A. Sharma. Prudence in vacillatory language identification. *Mathematical Systems Theory*, 28(3):267–279, May-June 1995.

[Kin77] E. Kinber. On a theory of inductive inference. Lecture Notes in Computer Science, 56, pages 435–440. Springer-Verlag, Berlin, 1977.

[Kir92] D. Kirsh. PDP learnability and innate knowledge of language. In S. Davis, editor, *Connectionism: Theory and Practice*, pages 297–322. Oxford University Press, NY, 1992.

[KLHM93] S. Kapur, B. Lust, W. Harbert, and G. Martohardjono. Universal grammar and learnability theory: The case of binding domains and the 'subset principle'. In E. Reuland and W. Abraham, editors, *Knowledge and Language*, volume I, pages 185–216. Kluwer, 1993.

[KR88] S. Kurtz and J. Royer. Prudence in language learning. In D. Haussler and L. Pitt, editors, *Proceedings of the Workshop on Computational Learning Theory*, pages 143–156. Morgan Kaufmann Publishers, Inc., 1988.

[LW94] S. Lange and P. Watson. Machine discovery in the presence of incomplete or ambiguous data. In K. Jantke and S. Arikawa, editors, *Algorithmic Learning Theory*, volume 872 of *Lecture Notes in Artificial Intelligence*, pages 438–452. Springer-Verlag, Berlin, Reinhardsbrunn Castle, Germany, October 1994.

[LZ92] S. Lange and T. Zeugmann. Types of monotonic language learning and their characterization. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory, Pittsburgh, Pennsylvania*, pages 377–390. ACM Press, 1992.

[MB72] D. Moeser and A. Bregman. The role of reference in the acquisition of a miniature artificial language. *Journal of Verbal Learning and Verbal Behavior*, 11:759–769, 1972.

[MB73] D. Moeser and A. Bregman. Imagery and language acquisition. *Journal of Verbal Learning and Verbal Behavior*, 12:91–98, 1973.

[McD92] D. McDermott. Robot planning. *AI Magazine*, 13(2):55–79, 1992.

[McN66] D. McNeill. Developmental psycholinguistics. In F. Smith and G. A. Miller, editors, *The Genesis of Language*, pages 15–84. MIT Press, 1966.

[Men86] E. Mendelson. *Introduction to Mathematical Logic.* Brooks-Cole, San Francisco, third edition, 1986.

[Muk92] Y. Mukouchi. Characterization of finite identification. In *Proceedings of the Third International Workshop on Analogical and Inductive Inference, Dagstuhl Castle, Germany*, pages 260–267, October 1992.

[MW87] R. Manzini and K. Wexler. Parameters, binding theory and learnability. *Linguistic Inquiry*, 18:413–444, 1987.

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

[OSW82] D. Osherson, M. Stob, and S. Weinstein. Ideal learning machines. *Cognitive Science*, 6:277–290, 1982.

[OSW83] D. Osherson, M. Stob, and S. Weinstein. Note on a central lemma of learning theory. *Journal of Mathematical Psychology*, 27:86–92, 1983.

[OSW84] D. Osherson, M. Stob, and S. Weinstein. Learning theory and natural language. *Cognition*, 17:1–28, 1984.

[OSW86a] D. Osherson, M. Stob, and S. Weinstein. An analysis of a learning paradigm. In W. Demopoulos and A. Marras, editors, *Language Learning and Concept Acquisition.* Ablex Publ. Co., 1986.

[OSW86b] 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.

[OSW86c] 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.

[OW82a] D. Osherson and S. Weinstein. Criteria for language learning. *Information and Control*, 52:123–138, 1982.

[OW82b] D. Osherson and S. Weinstein. A note on formal learning theory. *Cognition*, 11:77–88, 1982.

[Pen89] R. Penrose. *The Emperor's New Mind.* Oxford University Press, NY, 1989.

[Pen94] R. Penrose. *Shadows of the Mind.* Oxford University Press, NY, 1994.

[PH77] J. Paris and L. Harrington. A mathematical incompleteness in Peano arithmetic. In J. Barwise, editor, *Handbook of Mathematical Logic.* North Holland, 1977.

[Pin79] S. Pinker. Formal models of language learning. *Cognition*, 7:217–283, 1979.

[Pyl84] Z. Pylyshyn. *Computation and Cognition: Toward A Foundation For Cognitive Science.* MIT , Cambridge, MA, 1984.

[RC94] J. Royer and J. Case. *Subrecursive Programming Systems: Complexity and Succinctness.* Progress in Theoretical Computer Science. Birkhäuser Boston, 1994.

[Ric80] G. Riccardi. *The Independence of Control Structures in Abstract Programming Systems.* PhD thesis, State University of New York at Buffalo, 1980.

[Ric81] G. Riccardi. The independence of control structures in abstract programming systems. *Journal of Computer and System Sciences*, 22:107–143, 1981.

[Rog58] H. Rogers. Gödel numberings of partial recursive functions. *Journal of Symbolic Logic*, 23:331–341, 1958.

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

[Roy87] J. Royer. *A Connotational Theory of Program Structure.* Lecture Notes in Computer Science 273. Springer Verlag, 1987.

[Sim85] S. Simpson. Nonprovability of certain combinatorial properties of finite trees. In L. Harrington, M. Morley, A. Schedrov, and S. Simpson, editors, *Harvey Friedman's Research on the Foundations of Mathematics*, pages 87–117. North Holland, 1985.

[Sim87] S. Simpson. Unprovable theorems and fast–growing functions. In S. Simpson, editor, *Logic and Combinatorics*, AMS Comtemporary Mathematics, pages 359–394. 1987.

[Smi94] C. Smith. *A Recursive Introduction to the Theory of Computation.* Springer-Verlag, New York, 1994.

[Smu61] R. Smullyan. *Theory of Formal Systems.* Annals of Mathematics Studies, No. 47. 1961.

[Soa87] R. Soare. *Recursively Enumerable Sets and Degrees.* Springer-Verlag, 1987.

[Spe94] E. Spelke. Initial knowledge: Six suggestions. *Cognition*, 50:431–445, 1994.

[SR84] G. Schäfer-Richter. *Uber Eingabeabhangigkeit und Komplexitat von Inferenzstrategien.* PhD thesis, RWTH Aachen, 1984.

[TM87] T. Toffoli and N. Margolus. *Cellular Automata Machines.* MIT Press, 1987.

[Tof77] T. Toffoli. Cellular automata machines. Technical Report 208, Comp. Comm. Sci. Dept., University of Michigan, 1977.

[WC80] K. Wexler and P. Culicover. *Formal Principles of Language Acquisition.* MIT Press, Cambridge, Mass, 1980.

[Wex82] K. Wexler. On extensional learnability. *Cognition*, 11:89–95, 1982.

[Wex93] K. Wexler. The subset principle is an intensional principle. In E. Reuland and W. Abraham, editors, *Knowledge and Language*, volume I, pages 217–239. Kluwer, 1993.

[Wie77] R. Wiehagen. Identification of formal languages. Lecture Notes in Computer Science, 53, pages 571–579. Springer, 1977.

[WZ92] R. Wiehagen and T. Zeugmann. Too much information can be too much for learning efficiently. In K. Jantke, editor, *Proceedings of the Third International Workshop on Analogical and Inductive Inference*, volume 642 of *Lecture Notes in Artificial Intelligence*, pages 72–86. Springer-Verlag, Berlin, Dagstuhl Castle, Germany, October 1992.

[You73] P. Young. Easy constructions in complexity theory: Gap and speed-up theorems. *Proceedings of the AMS*, 37:555–563, 1973.

[Zus69] K. Zuse. *Rechnender Raum.* Vieweg, Braunshweig, 1969. Translated as *Calculating Space*, Tech. Transl. AZT–70–164–GEMIT, MIT Project MAC, 1970.