# Open Problems in *Systems that Learn*


Mark Fulk [1]

Department of Computer Science

University of Rochester

Rochester, New York 14627

email: fulk@cs.rochester.edu

Sanjay Jain

Department of Computer and Information Sciences

University of Delaware

Newark, Delaware 19716

email: sjain@cis.udel.edu

Daniel N. Osherson [2]

IDIAP

C. P. 609

CH-1920 Martigny

Switzerland

email: osherson@idiap.ch


March 12, 2007


[1]Supported by NSF grant CCR 832-0136.

[2]Supported in part by ONR grants N00014-87-K-0401 and N00014-89-J-1725.

## Abstract

In this paper we solve some of the open problems in [19]. We also give partial solutions to some other open problems in the book. In particular we show that the collection of classes of languages that can be identified on "noisy" text (i.e. a text which may contain some elements which are not in the language being learned) strictly contains the collection of classes of languages that can be identified on "imperfect" text (i.e. a text which may contain some extra elements and may leave out some elements from the language being learned). We also show that memory limited identification is strictly more restrictive that memory bounded identification. Besides solving the above two open problems from [19] we also give partial solutions to other open problems in [19].

# 1 Introduction

A typical learning scenario involving a subject learning a concept could be described thus: Subject receives successive finite pieces of data, about the concept being learned, over time. Based upon these data, the subject, each time, either holds or changes its previous explanation for the concept. The subject *converges* to a particular explanation, if, after some time, it always holds that explanation. The subject is said to *learn* the concept, just in case, it converges to a correct explanation for the concept. Computational learning theory provides a framework for studying problems of this nature when the subject is an algorithmic device. Instances of learning situations are inductive hypothesis formation and language acquisition.

The following is based on the theme of inductive inference studied by Gold [13]. Picture a scientist performing all possible experiments (in arbitrary order) associated with a phenomenon, noting the result of each experiment, while simultaneously, but algorithmically, conjecturing a succession of candidate explanations for the phenomenon. A criterion of success is that the scientist converges to an explanation which correctly predicts the results of every experiment about the phenomenon. The set of all pairs of the form (experiment, corresponding result) associated with the phenomenon can be taken to be coded by a function from $N$ to $N$, where $N$ is the set of natural numbers. If the scientist in the above scenario is replaced by a machine (we call such a machine an *inductive inference machine* or IIM for short), then algorithmic identification in the limit of a program for a recursive function from its graph serves as a plausible model for the practice of science. A machine $\mathbf{M}$ **Ex**-*identifies* a function iff (by definition) the scientist is replaced by machine $\mathbf{M}$ in the above scenario for success. Given an IIM $\mathbf{M}$, let $\mathbf{Ex}(\mathbf{M})$ denote the class of functions **Ex**-identified by $\mathbf{M}$. **Ex** is defined to be the collection of classes $\mathcal{S}$, of recursive functions, such that, for some $\mathbf{M}$, $\mathcal{S} \subseteq \mathbf{Ex}(\mathbf{M})$.

A related idea to "scientific" inference of functions is Gold's seminal notion of *language identification* [13]. We will refer to it as **TxtEx**-*identification* following [6]. In the following, a language is a *recursively enumerable* (r.e.) set, and a *grammar* (type 0) for a language is a program that enumerates the language [14] in some fixed acceptable programming system [21, 22, 18].

According to Gold's paradigm, a child (modeled as a machine) receives (in arbitrary order) all and only the well-defined strings of a language (a *text* for the language), and simultaneously, conjectures a succession of candidate grammars for the language being received. A criterion of success is for the child to *converge* to a correct grammar for the language. A machine $\mathbf{M}$ **TxtEx**-*identifies* a language iff (by definition) the child is replaced by machine $\mathbf{M}$ in the above scenario for success. Given an IIM $\mathbf{M}$, let $\mathbf{TxtEx}(\mathbf{M})$ denote the class of languages **TxtEx**-identified by $\mathbf{M}$. **TxtEx** is defined to be the collection of classes $\mathcal{L}$, of r.e. languages, such that, for some $\mathbf{M}$, $\mathcal{L} \subseteq \mathbf{TxtEx}(\mathbf{M})$.

In this paper we consider some restrictions on the above model of learning. These restrictions were first considered by Osherson, Stob and Weinstein [19]. We show that the collection of classes of languages that can be **TxtEx**-identified on "noisy" text (i.e. a text which may contain some elements which are not in the language being learned) strictly contains the collection of classes of languages that can be **TxtEx**-identified on "imperfect" text (i.e. a text which may contain some extra elements and may leave out some elements from the language being learned). This solves an open question (open question 5.4.3A, page 105) in [19]. In Section 4 we show that memory limited identification is strictly more restrictive than memory bounded identification (see Section 4 for Definitions), which solves another open question (open question 4.4.1A, page 73) in [19]. We also give partial solutions to other open problems in [19]. Some of these results were announced in [16].

## 2 Preliminaries

### 2.1 Notations

Any unexplained recursion theoretic notation is from [22]. $N$ denotes the set of natural numbers, $\{0, 1, 2, 3, \ldots\}$. $e, i, j, k, l, m, n, p, q, r, s, x, y, z$, with or without decorations (decorations are subscript, superscirpt and the like), range over $N$. $*$ denotes a non-member of $N$ and is assumed to satisfy $(\forall n)[n < * < \infty]$. $a, b$ and $c$, with or without decorations, range over $(N \cup \{*\})$. $\emptyset$ denotes the empty set. $\in$, $\subseteq$, $\subset$, $\supseteq$, $\supset$ respectively denote member of, subset, proper subset, superset and proper superset. $\uparrow$ denotes undefined. $\downarrow$ denotes defined. $S$, with or without decorations, ranges over subsets of $N$. $S_1 \oplus S_2$ denotes the symmetric difference of the sets $S_1$ and $S_2$, i.e., $(S_1 - S_2) \cup (S_2 - S_1)$. $\mathrm{card}(S)$ denotes the cardinality of the set $S$. $\max(), \min()$ denote the maximum and minimum of a set respectively. For $n \in N$ and any two sets $S_1$ and $S_2$, $S_1 =^n S_2$ means $\text{card}(S_1 \oplus S_2) \leq n$; $S_1 =^* S_2$ means $\text{card}(S_1 \oplus S_2)$ is finite. $\mu x[Q(x)]$ is the least natural number $x$ such that $Q(x)$ is true (if such an $x$ exists).

$\eta$ and $\theta$ range over *partial* functions with arguments and values from $N$. $f$ and $g$, with or without decorations, range over *total* functions with arguments and values from $N$. For $n \in N$ and partial functions $\eta$ and $\theta$, $\eta =^n \theta$ means that $\text{card}(\{x \mid \eta(x) \neq \theta(x)\}) \leq n$; $\eta =^* \theta$ means that $\text{card}(\{x \mid \eta(x) \neq \theta(x)\})$ is finite. $\text{domain}(\eta)$ and $\text{range}(\eta)$ denote the domain and range of the function $\eta$, respectively.

$L$, with or without decorations, ranges over subsets of $N$, usually construed as a *language*. $\mathcal{E}$ denotes the class of all *recursively enumerable* languages. $\mathcal{L}$, with or without decorations, ranges over subsets of $\mathcal{E}$. $\overline{L}$ denotes the complement of $L$, i.e., $\overline{L} = N - L$.

$\varphi$ denotes a standard *acceptable* programming system [21, 22, 18]. $\Phi$ denotes an arbitrary Blum complexity measure [4, 14] for the $\varphi$-system. $\varphi_i$ denotes the partial computable function computed by program $i$ in the $\varphi$-system. $W_i = \text{domain}(\varphi_i)$. $W_i^s = \{x \leq s \mid \Phi_i(x) \leq s\}$. The set of all total recursive functions of one variable is denoted by $\mathcal{R}$. $\mathcal{S}, \mathcal{C}$, with or without decorations, range over subsets of $\mathcal{R}$. $\langle i, j\rangle$ stands for an arbitrary computable one to one encoding of all pairs of natural numbers onto $N$ [22] (we assume that $\langle i, j\rangle \geq \max(\{i, j\})$). $\pi_1$ and $\pi_2$ are the corresponding projection functions. For $n > 2$, $\langle \cdot, \cdot \rangle$ is extended to $n$-tuples in the usual way. We let $\langle\ \rangle = 0$ and $\langle x \rangle = x$. $S \subseteq N$ is called single-valued just in case $\{(x, y) \mid \langle x, y\rangle \in S\}$ represents a function. A single-valued set is said to be single-valued total (*svt*), just in case, the function it represents is total.

The quantifiers '$\overset{\infty}{\forall}$' and '$\overset{\infty}{\exists}$', essentially from [4], mean 'for all but finitely many' and 'there exist infinitely many', respectively. The quantifier '$\exists!$' denotes 'there exists a unique'.

## 2.2 Fundamental Learning Paradigms

In this section we briefly discuss notions from the machine learning theoretic literature. For detailed discussion see [19, 7, 13, 1, 17, 3].

A *text* is a mapping from $N$ into $(N \cup \{\#\})$. Texts are also referred to as *information sequences*. A *segment* (also called finite *sequence*) is a mapping, for some natural number $i$, from $\{x \mid x < i\}$ into $(N \cup \{\#\})$. $\Lambda$ denotes an empty sequence. For notational convenience, we sometimes write a sequence, $\{(0, x_0), (1, x_1), \ldots, (k, x_k)\}$ as simply $(x_0, x_1, \ldots, x_k)$. $T$, with or without decorations, ranges over texts. $\sigma, \tau$, with or without decorations, range over segments. $|\sigma|$ denotes the length of $\sigma$, i.e., the number of elements in $\sigma$. $\sigma_1 \diamond \sigma_2$ denotes the concatenation of $\sigma_1$ and $\sigma_2$, i.e., if $\sigma = \sigma_1 \diamond \sigma_2$, then, for all $x$,

$$\sigma(x) = \begin{cases} \sigma_1(x), & \text{if } x < |\sigma_1|; \\ \sigma_2(x - |\sigma_1|), & \text{if } |\sigma_1| \leq x < |\sigma_1| + |\sigma_2|; \\ \uparrow, & \text{otherwise.} \end{cases}$$

$T[n]$ denotes the initial segment of $T$ with length $n$. $\text{content}(T) = \text{range}(T) - \{\#\}$; intuitively it is the set of meaningful things presented in $T$. $\text{content}(\sigma) = \text{range}(\sigma) - \{\#\}$. A *text* for a language $L$ is an information sequence $T$ such that $\text{content}(T) = L$.

$\mathbf{M}$, with or without decorations, ranges over IIMs. Inductive Inference Machines have been used in the study of identification of recursive functions as well as recursively enumerable languages [13, 7, 6, 1, 17, 9, 3, 19]. $\mathbf{M}(\sigma)$ denotes the last output of $\mathbf{M}$, if any, by the time it has received $\sigma$ as input. Without loss of generality, we will assume that $\mathbf{M}(\sigma)$ is always defined. $\mathbf{M}(T){\downarrow} = i$ iff $(\stackrel{\infty}{\forall} n)[\mathbf{M}(T[n]) = i]$. We write $\mathbf{M}(T){\downarrow}$ iff $(\exists i)[\mathbf{M}(T){\downarrow} = i]$.

For a total function $f$, $f[n]$ denotes the sequence $((0, f(0)), (1, f(1)), \ldots, (n-1, f(n-1)))$. For function learning the input sequence given to the IIM is $(0, f(0)), (1, f(1)), \ldots$, where $f$ is the function being learned. $\mathbf{M}(f[n])$ denotes the last output of $\mathbf{M}$, if any, by the time it has received $f[n]$ as input. Without loss of generality, we will assume that $\mathbf{M}(f[n])$ is always defined. $\mathbf{M}(f){\downarrow} = i$ iff $(\stackrel{\infty}{\forall} n)[\mathbf{M}(f[n]) = i]$. We write $\mathbf{M}(f){\downarrow}$ iff $(\exists i)[\mathbf{M}(f){\downarrow} = i]$. We now formally define $\mathbf{Ex}$-identification introduced in Section 1.

**Definition 1** [13, 3, 7] Recall that $a$ ranges over $N \cup \{*\}$.

(a) $\mathbf{M}$ $\mathbf{Ex}^a$-*identifies* $f$ iff both $\mathbf{M}(f){\downarrow}$ and $\varphi_{\mathbf{M}(f)} =^a f$. If $\mathbf{M}$ $\mathbf{Ex}^a$-identifies $f$, then we write $f \in \mathbf{Ex}^a(\mathbf{M})$.

(b) $\mathbf{Ex}^a = \{\mathcal{S} \subseteq \mathcal{R} \mid (\exists \mathbf{M})[\mathcal{S} \subseteq \mathbf{Ex}^a(\mathbf{M})]\}$.

In the above definitions $a$ stands for the number of anomalies allowed in the final program. $a = *$ means that unbounded but finite number of anomalies is allowed in the final program.

A criterion of success for language learning can be defined similarly.

**Definition 2** [13, 6]

(a) $\mathbf{M}$ $\mathbf{TxtEx}^a$-*identifies* $L$ iff $(\forall$ texts $T$ for $L)[\mathbf{M}(T)\downarrow \wedge W_{\mathbf{M}(T)} =^a L]$. If $\mathbf{M}$ $\mathbf{TxtEx}^a$-identifies $L$, then we write $L \in \mathbf{TxtEx}^a(\mathbf{M})$.

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

We now introduce some technical notions which are useful in the study of learning capabilities of the IIMs.

**Definition 3**

(a) [9] $\sigma$ is a $\mathbf{TxtEx}$-*stabilizing segment for* $\mathbf{M}$ *on* $L$ iff $\mathrm{content}(\sigma) \subseteq L$ and $(\forall \sigma' \mid \mathrm{content}(\sigma') \subseteq L \wedge \sigma \subseteq \sigma')[\mathbf{M}(\sigma') = \mathbf{M}(\sigma)]$.

(b) [3, 20] $\sigma$ is a $\mathbf{TxtEx}^a$-*locking sequence for* $\mathbf{M}$ *on* $L$ iff $\sigma$ is a $\mathbf{TxtEx}$-stabilizing segment for $\mathbf{M}$ on $L$ and $W_{\mathbf{M}(\sigma)} =^a L$.

We often refer to $\mathbf{TxtEx}^a$-locking sequence by just locking sequence ($a$ will be clear from context). We now present a very important lemma in learning theory due to L. Blum and M. Blum.

**Lemma 4** [3, 20] *If* $\mathbf{M}$ $\mathbf{TxtEx}^a$*-identifies* $L$*, then there is a* $\mathbf{TxtEx}^a$*-locking sequence for* $\mathbf{M}$ *on* $L$*.*

Case and Smith [7] introduced another infinite hierarchy of identification criteria which we describe below. "**Bc**" stands for *behaviorally correct*. Barzdin [2] independently introduced a similar notion.

**Definition 5** [7]

(a) $\mathbf{M}$ $\mathbf{Bc}^a$-identifies $f$ iff $(\overset{\infty}{\forall} n)[\varphi_{\mathbf{M}(f[n])} =^a f]$. If $\mathbf{M}$ $\mathbf{Bc}^a$-identifies $f$, then we write $f \in \mathbf{Bc}^a(\mathbf{M})$.

(b) $\mathbf{Bc}^a = \{\mathcal{S} \subseteq \mathcal{R} \mid (\exists \mathbf{M})[\mathcal{S} \subseteq \mathbf{Bc}^a(\mathbf{M})]\}$.

**Definition 6** [6]

(a) $\mathbf{M}$ $\mathbf{TxtBc}^a$-identifies $L$ iff $(\forall$ texts $T$ for $L)(\overset{\infty}{\forall} n)[W_{\mathbf{M}(T[n])} =^a L]$. If $\mathbf{M}$ $\mathbf{TxtBc}^a$-identifies $L$, then we write $L \in \mathbf{TxtBc}^a(\mathbf{M})$.

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

We usually write $\mathbf{Ex}$ for $\mathbf{Ex}^0$, $\mathbf{TxtEx}$ for $\mathbf{TxtEx}^0$, $\mathbf{Bc}$ for $\mathbf{Bc}^0$, and $\mathbf{TxtBc}$ for $\mathbf{TxtBc}^0$.

# 3 Advantages of Having Noisy Text as Compared to Imperfect Text

In the real world input data is rarely free of error. Osherson, Stob and Weinstein considered three types of inaccuracies in input data.

(1) The input text may contain elements not in the language (noisy text).

(2) Some elements of the language may be absent from the text (incomplete text).

(3) A combination of (1) and (2) may occur (imperfect text).

They showed that inaccurate input restricts the learning capabilities of an inductive inference machine. They left open whether imperfect text commits strictly more harm than noisy text. We show that this is indeed the case.

**Definition 7** [19, 11]

(a) We say that a text $T$ is $a$-*noisy* for $L$ iff

(i) $L \subseteq \text{content}(T)$, and

(ii) $\text{card}(\text{content}(T) - L) \leq a$.

(b) We say that a text $T$ is $a$-*incomplete* for $L$ iff

(i) $\text{content}(T) \subseteq L$, and

(ii) $\text{card}(L - \text{content}(T)) \leq a$.

(c) We say that a text $T$ is $a$-*imperfect* for $L$ iff $\text{card}(L \oplus \text{content}(T)) \leq a$.

We now consider criteria for identification with respect to noisy, incomplete or imperfect text.

**Definition 8** Let $L \in \mathcal{E}$.

(a) We say that $\mathbf{M}$ $\mathbf{N}^a\mathbf{TxtEx}^b$-*identifies* $L$ iff $(\forall a$-noisy texts $T$ for $L)[\mathbf{M}(T)\downarrow \wedge W_{\mathbf{M}(T)} =^b L]$.

(b) We say that $\mathbf{M}$ $\mathbf{In}^a\mathbf{TxtEx}^b$-*identifies* $L$ iff $(\forall a$-incomplete texts $T$ for $L)[\mathbf{M}(T)\downarrow \wedge W_{\mathbf{M}(T)} =^b L]$.

(c) We say that $\mathbf{M}$ $\mathbf{Im}^a\mathbf{TxtEx}^b$-*identifies* $L$ iff $(\forall a$-imperfect texts $T$ for $L)[\mathbf{M}(T)\downarrow \wedge W_{\mathbf{M}(T)} =^b L]$.

In Definition 8(a), if $\mathbf{M}$ $\mathbf{N}^a\mathbf{TxtEx}^b$-identifies $L$, then we write $L \in \mathbf{N}^a\mathbf{TxtEx}^b(\mathbf{M})$. We denote by $\mathbf{N}^a\mathbf{TxtEx}^b$ the collection of language classes $\mathcal{L}$ such that, for some $\mathbf{M}$, $\mathcal{L} \subseteq \mathbf{N}^a\mathbf{TxtEx}^b(\mathbf{M})$. We do similarly for $\mathbf{In}^a\mathbf{TxtEx}^b$ and $\mathbf{Im}^a\mathbf{TxtEx}^b$.

We similarly define the criteria of identification for function inference. For a function $f$, let $L_f = \{\langle x, f(x)\rangle \mid x \in N\}$.

**Definition 9** Let $f \in \mathcal{R}$.

(a) We say that $\mathbf{M}$ $\mathbf{N}^a\mathbf{Ex}^b$-*identifies* $f$ iff $(\forall a$-noisy texts $T$ for $L_f)[\mathbf{M}(T)\downarrow \wedge \varphi_{\mathbf{M}(T)} =^b f]$.

(b) We say that $\mathbf{M}$ $\mathbf{In}^a\mathbf{Ex}^b$-*identifies* $f$ iff $(\forall a$-incomplete texts $T$ for $L_f)[\mathbf{M}(T)\downarrow \wedge \varphi_{\mathbf{M}(T)} =^b f]$.

(c) We say that $\mathbf{M}$ $\mathbf{Im}^a\mathbf{Ex}^b$-*identifies* $f$ iff $(\forall a$-imperfect texts $T$ for $L_f)[\mathbf{M}(T)\downarrow \wedge \varphi_{\mathbf{M}(T)} =^b f]$.

In Definition 9(a), if $\mathbf{M}$ $\mathbf{N}^a\mathbf{Ex}^b$-identifies $f$, then we write $f \in \mathbf{N}^a\mathbf{Ex}^b(\mathbf{M})$. We denote by $\mathbf{N}^a\mathbf{Ex}^b$ the collection of classes $\mathcal{S}$ such that for some $\mathbf{M}$, $\mathcal{S} \subseteq \mathbf{N}^a\mathbf{Ex}^b(\mathbf{M})$. We do similarly for $\mathbf{In}^a\mathbf{Ex}^b$ and $\mathbf{Im}^a\mathbf{Ex}^b$.

**Theorem 10** $\mathbf{N}^*\mathbf{Ex} - \mathbf{In}^1\mathbf{Ex}^* \neq \emptyset$.

As a corollary we obtain.

**Corollary 11** $\mathbf{Im}^*\mathbf{TxtEx} \subset \mathbf{N}^*\mathbf{TxtEx}$.

Proof of Theorem 10. Given $f \in \mathcal{R}$ define $f'$ as follows.

Let $i = \min(\{j \mid \varphi_j = f\})$. For $j < i$, let $err_j = \min(\{x \mid \varphi_j(x) \neq f(x)\})$.

$$f'(0) = \langle i, \langle err_0, err_1, err_2, \ldots, err_{i-1}\rangle\rangle.$$

$$\text{For all } j, k : f'(1 + \langle j, k\rangle) = f(j).$$

Let $\mathcal{C} = \{f' \mid f \in \mathcal{R}\}$. We show that $\mathcal{C} \in \mathbf{N}^*\mathbf{Ex} - \mathbf{In}^1\mathbf{Ex}^*$. Intuitively, for $f'$ defined as above, $f'(0)$ codes sufficient information about $f'$, so that $\mathcal{C}$ can be identified from noisy texts; however, if $f'(0)$ is missing from the input data, then identifying, $\mathcal{C}$ is equivalent to identifying $\mathcal{R}$.

**Claim 12** $\mathcal{C} \notin \mathbf{In}^1\mathbf{Ex}^*$.

Proof. Suppose by way of contradiction that $\mathbf{M}$ $\mathbf{In}^1\mathbf{Ex}^*$-identifies $\mathcal{C}$. We then describe a machine $\mathbf{M}'$ which $\mathbf{Ex}^*$-identifies $\mathcal{R}$. Given $f \in \mathcal{R}$, define a text $T'$ as follows. Let $T'(\langle x, k\rangle) = \langle 1 + \langle x, k\rangle, f(x)\rangle$. Note that $T'$ is a 1-incomplete text for $f'$. Also note that $T'[n]$ can be effectively constructed from $f[n]$. Let $G$ be a recursive function such that, for all $x, p$, $\varphi_{G(p)}(x) = \varphi_p(\langle 1 + \langle x, 0\rangle\rangle)$. Define $\mathbf{M}'$ as follows: $\mathbf{M}'(f[n]) = G(\mathbf{M}(T'[n]))$. Note that such a machine $\mathbf{M}'$ can easily be constructed from $\mathbf{M}$. Clearly, for $f \in \mathcal{R}$, if $\varphi_p =^* f'$, then $\varphi_{G(p)} =^* f$. Since $\mathbf{M}$ $\mathbf{In}^1\mathbf{Ex}^*$-identifies $\mathcal{C}$, it follows that $\mathbf{M}'$ $\mathbf{Ex}^*$-identifies $\mathcal{R}$. [3, 7] observed that $\mathcal{R} \notin \mathbf{Ex}^*$. Thus we have that $\mathcal{C} \notin \mathbf{In}^1\mathbf{Ex}^*$. □ (Claim 12)

**Claim 13** $\mathcal{C} \in \mathbf{N}^*\mathbf{Ex}$.

Proof. We describe a machine $\mathbf{M}$ which $\mathbf{N}^*\mathbf{Ex}$-identifies $\mathcal{C}$. Let $G$ be a recursive function such that, for all $e, z, j, k$,

$$\varphi_{G(e,z)}(0) = z;$$
$$\varphi_{G(e,z)}(1 + \langle j, k\rangle) = \varphi_e(j).$$

Suppose $f \in \mathcal{R}$ and $T$ is a $*$-noisy text for $f'$ ($\in \mathcal{C}$). We describe how $\mathbf{M}$ computes its output on $T[n]$. For this we first describe, $X_n, Y_n, e_n$ (which depend on $T, n$). Let

$$X_n = \{x \mid \langle 0, x\rangle \in \text{content}(T[n])\}$$

and

$$Y_n = \{(j,y) \mid (\exists k)[\langle 1+\langle j,k\rangle, y\rangle \in \text{content}(T[n]) \wedge$$
$$(\forall k' \mid (\exists y')[\langle 1{+}\langle j,k'\rangle, y'\rangle \in \text{content}(T[n])])(\exists k \geq k')[\langle 1{+}\langle j,k\rangle, y\rangle \in \text{content}(T[n])]]\}$$

Note that, for large enough $n$, $Y_n \subseteq f$. Let

$$e_n = \max(\{i \mid (\exists err_0, err_1, \ldots, err_{i-1} \mid \langle i, \langle err_0, err_1, err_2, \ldots err_{i-1}\rangle\rangle \in X_n)$$
$$[(\forall j < i)[\Phi_j(err_j) > n \vee (err_j, \varphi_j(err_j)) \notin Y_n]]\}).$$

It is easy to see that, for large enough $n$, $e_n = \min(\{j \mid \varphi_j = f\})$.

Let

$$z_n = \langle e_n, \langle err_0, \ldots, err_{e_n - 1}\rangle\rangle,$$

where, for $j < e_n$, $err_j = \min(\{n\} \cup \{x < n \mid \Phi_j(x) > n \vee \Phi_{e_n}(x) > n \vee \varphi_j(x) \neq \varphi_{e_n}(x)\})$. From, the definition of $f'$, it follows that for large enough $n$, $z_n = f'(0)$.

Let $\mathbf{M}(T[n]) = G(e_n, z_n)$. It is easy to see that, $f' \in \mathbf{N}^*\mathbf{Ex}(\mathbf{M})$. Since $f'$ was an arbitrary member of $\mathcal{C}$, we have $\mathcal{C} \subseteq \mathbf{N}^*\mathbf{Ex}(\mathbf{M})$. □ (Claim 13) ▮ (Theorem 10)

In [11, 15] it is shown that $\mathbf{In}^a\mathbf{Ex}^b \subseteq \mathbf{N}^a\mathbf{Ex}^b$. However, in the case of language learning, it is shown in [11, 15] that $\mathbf{In}^*\mathbf{TxtEx} - \mathbf{N}^*\mathbf{TxtEx}^* \neq \emptyset$. For further results relating different criteria of inference formed using inaccurate texts see [11, 15].

# 4 Memory Limited Identification Versus Memory Bounded Identification

In Gold's model of learning the learner is allowed to look at the *whole* initial segment of the text for its new conjecture. However a child, having a finite head size, cannot retain in its memory all the sentences it has heard. Motivated by this observation we consider the following learning criteria.

Let $\sigma^{-}i$ denote the last $i$ elements of $\sigma$ (in order). Formally, for all $x$,

$$\sigma^{-}i(x) = \begin{cases} \sigma(|\sigma| - i + x), & \text{if } x < i; \\ \uparrow, & \text{otherwise.} \end{cases}$$

Let $\sigma^{-}$ denote $\sigma$ with the last element removed. Formally, for all $x$,

$$\sigma^{-}(x) = \begin{cases} \sigma(x), & \text{if } x < |\sigma| - 1; \\ \uparrow, & \text{otherwise.} \end{cases}$$

**Definition 14** [19] $\mathbf{M}$ is *i-memory limited* iff $(\forall \sigma, \tau)[[\mathbf{M}(\sigma^-) = \mathbf{M}(\tau^-) \wedge \sigma^- i = \tau^- i] \Rightarrow \mathbf{M}(\sigma) = \mathbf{M}(\tau)]$.

Intuitively, $i$-memory limitation allows the machine to remember only its last conjecture and the last $i$ elements of the input.

**Definition 15** [19]

(a) *mem*, a mapping from finite sequences to finite sequences, is an *i-memory function* iff, for all $\sigma$, content$(mem(\sigma)) \subseteq$ content$(\sigma)$, $length(mem(\sigma)) = i$ and content$(mem(\sigma))$ − content$(mem(\sigma^-)) \subseteq \{\sigma^- 1\}$.

(b) $\mathbf{M}$ is *i-memory bounded* iff there is an $i$-memory function *mem* such that, $(\forall \sigma, \tau)[[\mathbf{M}(\sigma^-) = \mathbf{M}(\tau^-) \wedge \sigma^- 1 = \tau^- 1 \wedge mem(\sigma) = mem(\tau)] \Rightarrow \mathbf{M}(\sigma) = \mathbf{M}(\tau)]$.

Intuitively, $i$-memory boundedness allows a machine to remember only its last conjecture, the last element of the input, and $i$ *selected* elements of the input.

It was shown in [19] that $i$-memory limited machines **TxtEx**-identify the same classes of languages as 1-memory limited machines. It is easy to see that 0-memory bounded machines **TxtEx**-identify the same classes of languages as 1-memory limited machines. Osherson, Stob and Weinstein left it open whether 1-memory bounded machines can **TxtEx**-identify a class of languages not **TxtEx**-identifiable by memory limited machines. We show that there is a class of languages which can be **TxtEx**-identified by a 1-memory bounded machine, but cannot be **TxtEx**-identified by any 1-memory limited machine.

**Definition 16** [19] $\mathbf{MLTxtEx} = \{\mathcal{L} \subseteq \mathcal{E} \mid (\exists 1\text{-memory limited machine } \mathbf{M})[\mathcal{L} \subseteq \mathbf{TxtEx}(\mathbf{M})]\}$.

**Definition 17** [19] $\mathbf{MB}^i\mathbf{TxtEx} = \{\mathcal{L} \subseteq \mathcal{E} \mid (\exists i\text{-memory bounded machine } \mathbf{M})[\mathcal{L} \subseteq \mathbf{TxtEx}(\mathbf{M})]\}$.

**Theorem 18** $\mathbf{MB}^1\mathbf{TxtEx} - \mathbf{MLTxtEx} \neq \emptyset$.

**Proof**: Let $L_0 = \{\langle 0, x\rangle \mid x \in N\}$; for $i \geq 1$, let $L_i = \{\langle 1, 0\rangle\} \cup \{\langle 0, x\rangle \mid x \leq i\}$. Let $\mathcal{L} = \{L_i \mid i \in N\}$.

We will show that $\mathcal{L} \in \mathbf{MB}^1\mathbf{TxtEx} - \mathbf{MLTxtEx}$. Intuitively, a memory bounded machine can remember the largest $x$ such that $\langle 0, x\rangle$ is in the input sequence, and thus **TxtEx**-identify

$\mathcal{L}$. However, a memory limited machine, which **TxtEx**-identifies $L_0$, cannot always remember the largest $x$ such that $\langle 0, x\rangle$ is in the input sequence.

**Claim 19** $\mathcal{L} \in \mathbf{MB}^1\mathbf{TxtEx}$.

**Proof**: Let $proj_2$ be a function, from finite sequences to $N$, such that, for all $x$, $proj_2((x)) = \pi_2(x)$. Let $mem$ be defined as follows.

$$mem(\Lambda) = (\#),$$

$$mem(\sigma \diamond (\#)) = mem(\sigma),$$

and

$$mem(\sigma \diamond (x)) = \begin{cases} (x), & \text{if } \sigma = \Lambda \vee mem(\sigma) = (\#) \vee \\ & [proj_2(mem(\sigma)) < \pi_2(x) \text{ and } \pi_1(x) = 0]; \\ mem(\sigma), & \text{otherwise.} \end{cases}$$

Let $g$ be a recursive function such that $W_{g(j)} = L_j$. Let $\lambda x, y.pad(x, y)$ be a 1-1 padding function for languages (thus for all $x, y$, $W_{pad(x,y)} = W_x$). Let $pad_2^{-1}$ be a right projection function for $pad$: $pad_2^{-1}(pad(x, y)) = y$.

Let $\mathbf{M}$ be an inductive inference machine such that:

$$\mathbf{M}(\Lambda) = pad(g(0), 0),$$

$$\mathbf{M}(\sigma \diamond (\#)) = \mathbf{M}(\sigma),$$

and

$$\mathbf{M}(\sigma \diamond (x)) = \begin{cases} pad(g(proj_2(mem(\sigma \diamond (x)))), 1), & \text{if } x = \langle 1, 0\rangle \vee pad_2^{-1}(\mathbf{M}(\sigma)) = 1; \\ pad(g(0), 0), & \text{otherwise.} \end{cases}$$

It is easy to see that $\mathbf{M}$ is 1-memory bounded (with the memory function $mem$) and $\mathcal{L} \subseteq \mathbf{TxtEx}(\mathbf{M})$. $\Box$ (Claim 19)

**Claim 20** $\mathcal{L} \notin \mathbf{MLTxtEx}$.

**Proof**: Suppose $\mathbf{M}$ is a 1-memory limited and $L_0 \in \mathbf{TxtEx}(\mathbf{M})$. We show that $\mathcal{L} \not\subseteq \mathbf{TxtEx}(\mathbf{M})$. By Lemma 4, there exists a locking sequence, $\sigma$, for $\mathbf{M}$ on $L_0$. Let $m = max(\{x \mid \langle 0, x\rangle \in \text{content}(\sigma)\})$. Let $\sigma'$ be an extension of $\sigma$ such that content$(\sigma') = \{\langle 0, x\rangle \mid x \leq m\}$.

Let $T = \sigma' \diamond (\langle 0, m+1 \rangle) \diamond (\langle 1, 0 \rangle) \diamond (\langle 1, 0 \rangle) \diamond (\langle 1, 0 \rangle) \ldots$ and $T' = \sigma' \diamond (\langle 1, 0 \rangle) \diamond (\langle 1, 0 \rangle) \diamond (\langle 1, 0 \rangle) \ldots$.

Since $\mathbf{M}$ is 1-memory limited and $\mathbf{M}(\sigma' \diamond (\langle 0, m+1 \rangle)) = \mathbf{M}(\sigma')$ we have $\mathbf{M}(T') = \mathbf{M}(T)$. But $T, T'$ are texts for different languages in $\mathcal{L}$. Thus $\mathcal{L} \not\subseteq \mathbf{TxtEx}(\mathbf{M})$. $\square$ (Claim 20) ▮ (Theorem 18)

A similar idea can be used to show that

**Theorem 21** $\mathbf{MB}^{i+1}\mathbf{TxtEx} - \mathbf{MB}^{i}\mathbf{TxtEx} \neq \emptyset$.

## 5 Decisiveness

In the traditional model of learning a learner may conjecture a theory which it has abandoned earlier. It may be reasonable to expect that the learner should not conjecture a theory it has once abandoned. A machine is said to be decisive if it never conjectures a grammar for a language which it has already abandoned. Formally

**Definition 22** [19] $\mathbf{M}$ is *decisive* if $(\forall T)(\forall n_1 < n_2 < n_3)[W_{\mathbf{M}(T[n_1])} \neq W_{\mathbf{M}(T[n_2])} \Rightarrow W_{\mathbf{M}(T[n_1])} \neq W_{\mathbf{M}(T[n_3])}]$.

It is open at present if decisiveness restricts **TxtEx** language learning (open question 4.5.5A, page 82 in [19]). It was shown independently by Schafer [23] and Fulk that decisiveness does not restrict **Ex** function learning. Fulk also showed that decisiveness does not restrict **Bc** function learning. We show that decisiveness restricts **TxtBc** language learning.

**Theorem 23** *There exists a language class $\mathcal{L} \in$ **TxtBc** which cannot be **TxtBc**-identified by any decisive machine.*

**Proof**: Let $\mathbf{M}_0, \mathbf{M}_1, \ldots$ be a recursive enumeration of all the IIMs. Let $L_j = \{\langle j, i \rangle \mid i \in N\}$, $\sigma_{j,k} = (\langle j, 0 \rangle, ..., \langle j, k \rangle)$, $L^1_{j,k} = \{\langle j, i \rangle \mid i \leq k\}$, and $L^2_{j,k} = W_{\mathbf{M}_j(\sigma_{j,k})}$. If $(\exists k)[L^1_{j,k} \subset L^2_{j,k} \subseteq L_j]$, then let $k_0$ be the least $k$ such that $[L^1_{j,k} \subset L^2_{j,k} \subseteq L_j]$ and then let $\chi_j = \{L^1_{j,k_0}, L^2_{j,k_0}\}$; otherwise, let $\chi_j = \{L_j\}$. Let $\mathcal{L} = \bigcup_j \chi_j$.

We will show that $\mathcal{L} \in \mathbf{TxtBc}$, and that no decisive machine can **TxtBc**-identify $\mathcal{L}$.

Informally, we show that, if $\neg(\exists k)[L^1_{j,k} \subset L^2_{j,k} \subseteq L_j]$, then $L_j \notin \mathbf{TxtEx}(\mathbf{M})$; if $L^1_{j,k} \subset L^2_{j,k} \subseteq L_j$, then either $\mathbf{M}_j$ is not decisive or it cannot **TxtEx**-identify both $L^1_{j,k}$ and $L^2_{j,k}$. Thus, either $\mathbf{M}_j$ is not decisive or $\chi_j \not\subseteq \mathbf{TxtEx}(\mathbf{M}_j)$.

Proof of $\mathcal{L} \in \mathbf{TxtBc}$ is based on utilization of the fact that, if $(\exists k)[L^1_{j,k} \subset L^2_{j,k} \subseteq L_j]$, then the least such $k$ can be found in the limit.

**Claim 24** $\mathcal{L} \in \mathbf{TxtBc}$.

**Proof**: Note that $L \in \chi_j \Rightarrow L \subseteq L_j$. Consider $\mathbf{M}$ which behaves as follows:

$\mathbf{M}$ on input $T[n]$

Let $j$ be such that content$(T[n]) \subseteq L_j$.

**if** $n \leq 1$

**then** let $\mathbf{M}(T[n])$ be a grammar for $L_j$.

**else**

let $Cand = \{k \leq n \mid L^1_{j,k} \subset W^n_{\mathbf{M}_j(\sigma_{j,k})} \subseteq L_j\}$.

**if** $Cand = \emptyset$

**then** let $\mathbf{M}(T[n])$ be a grammar for $L_j$.

**else**

let $k_0 = min(Cand)$;

**if** content$(T[n]) \subseteq L^1_{j,k_0}$

**then** let $\mathbf{M}(T[n]) = f(j, k_0)$, where $f$ is defined below.

**else** let $\mathbf{M}(T[n]) = g(j, k_0)$, where $g$ is defined below.

**endif**

**endif**

**endif**

End $\mathbf{M}$

In the above, $f$ and $g$ are such that:

$$W_{f(j,k)} = \begin{cases} L_j, & \text{if } W_{\mathbf{M}_j(\sigma_j,k)} \not\subseteq L_j; \\ L^1_{j,k}, & \text{otherwise.} \end{cases}$$

$$W_{g(j,k)} = \begin{cases} L_j, & \text{if } W_{\mathbf{M}_j(\sigma_j,k)} \not\subseteq L_j; \\ L^2_{j,k} \cap L_j, & \text{otherwise.} \end{cases}$$

We claim that $\mathbf{M}$ **TxtBc**-identifies $\mathcal{L}$. Let $T$ be a text for $L \in \chi_j$. Let $P(j,k)$ be the property that $L^1_{j,k} \subset L^2_{j,k} \subseteq L_j$. Now consider the following cases.

*Case 1*: $(\forall k)[\neg P(j,k)]$.

In this case $L = L_j$. We claim that if $\mathbf{M}$ outputs $f(j,k)$ or $g(j,k)$, then $[W_{f(j,k)} = W_{g(j,k)} = L_j]$. This will imply that $\mathbf{M}$ **TxtBc**-*identifies* $L$. Since $f(j,k)$ or $g(j,k)$ is output by $\mathbf{M}$ only if $W^n_{\mathbf{M}_j(\sigma_{j,k})} \supset L^1_{j,k}$, $[(\forall r)[\neg P(j,r)]$ and $[\mathbf{M}$ outputs $f(j,k)$ or $g(j,k)]]$ implies that $W_{\mathbf{M}_j(\sigma_{j,k})} \not\subseteq L_j$. Thus $W_{f(j,k)} = W_{g(j,k)} = L_j$.

*Case 2*: $(\exists k)[P(j,k)]$.

Let $k_0$ be the least $k$ such that $P(j,k)$. Now for large enough $n$, *Cand*, as computed by $\mathbf{M}$ on input $T[n]$, would contain $k_0$ as its least element. Now consider the following cases.

*Case 2.1*: $L = L^1_{j,k_0}$.

In this case, for sufficiently large $n$, $\mathbf{M}$, on input $T[n]$, outputs $f(j,k_0)$. Since $W_{\mathbf{M}_j(\sigma_{j,k_0})} \subseteq L_j$, $W_{f(j,k_0)} = L^1_{j,k_0} = L$.

*Case 2.2*: $L = L^2_{j,k_0}$.

In this case, for sufficiently large $n$, $\mathbf{M}$, on input $T[n]$, outputs $g(j,k_0)$. Since $W_{\mathbf{M}_j(\sigma_{j,k_0})} \subseteq L_j$, $W_{g(j,k_0)} = L^2_{j,k_0} = L$. $\square$ (Claim 24)

**Claim 25** *$\mathcal{L}$ is not* **TxtBc**-*identifiable by any decisive machine.*

**Proof**: Suppose by way of contradiction that machine $\mathbf{M}_j$ *decisively* **TxtBc** identifies $\mathcal{L}$.

Now consider $\chi_j$. Let $P(j,k)$ be the property that $L^1_{j,k} \subset L^2_{j,k} \subseteq L_j$.

If $(\forall k)[\neg P(j,k)]$, then $L_j \in \mathcal{L}$ which is not **TxtEx**-identified by $\mathbf{M}_j$.

If $(\exists k)[P(j,k)]$, then let $k_0$ be the least such $k$. Now $L^1_{j,k_0}, L^2_{j,k_0} \in \mathcal{L}$. Since on $\sigma_{j,k_0}$ $\mathbf{M}_j$ outputs a grammar for $L^2_{j,k_0} \neq L^1_{j,k_0}$, there must be extension $\sigma$ of $\sigma_{j,k_0}$ such that content$(\sigma) =$

$L^1_{j,k_0}$ and $W_{\mathbf{M}_j(\sigma)} = L^1_{j,k_0}$. Also there must be an extension $\sigma'$ of $\sigma$, such that content$(\sigma') \subseteq L^2_{j,k_0}$ and $W_{\mathbf{M}_j(\sigma')} = L^2_{j,k_0}$ (since $\mathbf{M}_j$ identifies both $L^1_{j,k_0}, L^2_{j,k_0}$). But then $\mathbf{M}_j$ is not decisive.

This proves the claim. $\square$ (Claim 25) ▮ (Theorem 23)

## 6 TxtFex and TxtFEXT Identification

In **TxtEx**-identification a machine is required to converge to a *single* grammar for the language it is learning. [5] and [19] consider the situation in which the requirement to converge to a single grammar is relaxed and the machine is allowed to vacillate between a finite number of (nearly) correct grammars.

**Definition 26** [5]

(a) $\mathbf{M}$ $\mathbf{TxtFex}^a_b$-*identifies* $L$ iff $(\forall$ texts $T$ for $L)(\exists S \mid \mathrm{card}(S) \leq b \wedge (\forall i \in S)[W_i =^a L])(\overset{\infty}{\forall} n)[\mathbf{M}(T[n]) \in S]$. If $\mathbf{M}$ $\mathbf{TxtFex}^a_b$-identifies $L$, then we write $L \in \mathbf{TxtFex}^a_b(\mathbf{M})$.

(b) $\mathbf{TxtFex}^a_b = \{\mathcal{L} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{TxtFex}^a_b(\mathbf{M})]\}$.

Osherson, Stob, and Weinstein referred to $\mathbf{TxtFex}^*_*$ as BFD identification.

**Definition 27** [19]

(a) $\mathbf{M}$ $\mathbf{TxtFEXT}^a_b$-*identifies* $L$ iff $(\forall$ texts $T$ for $L)(\exists S \mid \mathrm{card}(S) \leq b \wedge (\forall i, j \in S)[W_i = W_j =^a L])(\overset{\infty}{\forall} n)[\mathbf{M}(T[n]) \in S]$. If $\mathbf{M}$ $\mathbf{TxtFEXT}^a_b$-identifies $L$, then we write $L \in \mathbf{TxtFEXT}^a_b(\mathbf{M})$.

(b) $\mathbf{TxtFEXT}^a_b = \{\mathcal{L} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{TxtFEXT}^a_b(\mathbf{M})]\}$.

Osherson, Stob, and Weinstein left open whether $\mathbf{TxtFex}^*_* = \mathbf{TxtFEXT}^*_*$ (open question 6.5.3C, page 142 [19]). We give a partial solution to this problem.

**Theorem 28** $(\forall i, j)[\mathbf{TxtFex}^j_i \subseteq \mathbf{TxtFEXT}^{cj}_i]$ *where $c$ depends only on $i$.*

**Proof**: For any given $\mathbf{M}$ we construct $\mathbf{M}'$ such that if $\mathbf{TxtFex}^j_i(\mathbf{M}) \subseteq \mathbf{TxtFEXT}^{cj}_i(\mathbf{M}')$. The proof is based on a *careful* (partial) simulation of grammars output by $\mathbf{M}$.

$\mathbf{M}'$ on input $T[n]$:

Let $p = \mathbf{M}(T[n])$.

Let $l = \min(\{n\} \cup \{m < n \mid \mathrm{card}(\{\mathbf{M}(T[m']) \mid m \leq m' \leq n\}) \leq i\})$.

Let $S = \{\mathbf{M}(T[m']) \mid l \leq m' \leq n\}$.

(* Intuitively, $S$ is the set of the last $i$ distinct programs output by $\mathbf{M}$ on $T[n]$ (if the number of distinct programs output by $\mathbf{M}$ on $T[n]$ is $< i$, then $S$ is the set of all the programs output by $\mathbf{M}$ on the initial segments of $T[n]$) *)

Output $P(p, S - \{p\})$, where $P$ is a fixed recursive function such that

$W_{P(p,S)} = W_p \cup \bigcup\{ExtraOut(p, p_1, p_2, ..., p_r) \mid r \leq \mathrm{card}(S) \wedge (\forall k \mid 1 \leq k \leq r)[p_k \in S] \wedge (\forall k, k' \mid 1 \leq k < k' \leq r)[p_k \neq p_{k'}]\}$, where

$ExtraOut(p_0, p_1, p_2, ..p_r) = \bigcup\{W_{p_r}^{n_r} \mid (\exists n_0 \geq n_1 \geq n_2 \geq \ldots \geq n_r)(\forall i < r)[\mathrm{card}((W_{p_i}^{n_i} - W_{p_{i+1}}) \cup (W_{p_{i+1}}^{n_{i+1}} - W_{p_i}^{n_i})) \leq 2 * j]\}$.

End

We now argue that $\mathbf{TxtFex}_i^j(\mathbf{M}) \subseteq \mathbf{TxtFEXT}_i^{cj}(\mathbf{M}')$.

Let $T$ be any text for $L \in \mathbf{TxtFex}_i^j(\mathbf{M})$. Consider $\mathbf{M}'$ on $T$. Let $l = \min(\{m \mid \mathrm{card}(\{\mathbf{M}(T[m']) \mid m \leq m'\}) \leq i\})$. Let $S = \{\mathbf{M}(T[m'] \mid l \leq m'\}$. Intuitively, $S$ is the set of the last $i$ distinct grammars output by $\mathbf{M}$ on $T$ (if $\mathbf{M}$ on $T$ outputs less than $i$ distinct grammars, then $S$ is the set of distinct grammars output by $\mathbf{M}$ on $T$). Let $Q$ be the set of grammars which $\mathbf{M}$ outputs infinitely often on $T$. Clearly, $\{P(q, S - \{q\}) \mid q \in Q\}$ is the set of grammars which $\mathbf{M}'$ outputs infinitely often on $T$.

We need to show that, for each $q \in Q$, $W_{P(q,S-\{q\})} =^{cj} L$ and that all of the grammars in $\{P(q, Q - \{q\}) \mid q \in Q\}$, are for the same language.

Note that, for $q \in S$, for $p_1, p_2, \ldots, p_r$ such that $r \leq \mathrm{card}(S - \{q\}) \wedge (\forall k \mid 1 \leq k \leq r)[p_k \in S - \{q\}] \wedge (\forall k, k' \mid 1 \leq k < k' \leq r)[p_k \neq p_{k'}]\}$, $ExtraOut(q, p_1, p_2, \ldots, p_r) - W_q$ has atmost $2j \cdot r$ elements. Thus, for $q \in Q$, since $W_q =^j L$, $W_{P(q,S-\{q\})} =^{cj} L$, where $c$ depends only on $i$.

It remains to show that all the grammars in $\{P(q, Q - \{q\}) \mid q \in Q\}$, are for the same language. Assume without loss of generality that $\mathrm{card}(Q) \geq 2$ (otherwise we are done). Suppose $q_1, q_2 \in Q$, such that $q_1 \neq q_2$. We show that $W_{P(q_1,S-\{q_1\})} \subseteq W_{P(q_2,S-\{q_2\})}$. (Since $q_1, q_2$ are arbitrary members of $Q$, this suffices to show that all the grammars in $\{P(q, Q - \{q\}) \mid q \in$

$Q\}$, are for the same language). Clearly, $W_{P(q_2,S-\{q_2\})} \supseteq W_{q_1}$ (since $W_{q_1} =^{2j} W_{q_2}$, $W_{q_1} = \mathit{ExtraOut}(q_2,q_1) \subseteq W_{P(q_2,S-\{q_2\})}$). Now consider $\mathit{ExtraOut}(q_1,p_1,p_2,..,p_r)$ such that $r \leq \mathrm{card}(S-\{q_1\}) \wedge (\forall k \mid 1 \leq k \leq r)[p_k \in S-\{q_1\}] \wedge (\forall k,k' \mid 1 \leq k < k' \leq r)[p_k \neq p_{k'}]$.

*Case 1*: $q_2 = p_k$ for some $k$.

In this case clearly, $\mathit{ExtraOut}(q_2,p_{k+1},p_{k+2},\ldots,p_r) \supseteq \mathit{ExtraOut}(q_1,p_1,p_2,\ldots)$

*Case 2*: Not case 1.

In this case $\mathit{ExtraOut}(q_2,q_1,p_1,\ldots,p_r) = \mathit{ExtraOut}(q_1,p_1,p_2,..,p_m)$. This is so because $\mathrm{card}((W_{q_1}-W_{q_2}) \cup (W_{q_2}-W_{q_1})) \leq 2*j$.

From the above cases it follows that $W_{P(q_1,S-\{q_1\})} \subseteq W_{P(q_2,S-\{q_2\})}$. Thus $\mathbf{TxtFex}_i^j(\mathbf{M}) \subseteq \mathbf{TxtFEXT}_i^{cj}(\mathbf{M}')$. ∎ (Theorem 28)

Since the construction of $\mathbf{M}'$ in the above proof is effective in $\mathbf{M}$ and $i$ we also have

**Theorem 29** $(\forall j)[\mathbf{TxtFex}_*^j \subseteq \mathbf{TxtFEXT}_*^*]$.

# 7 Conclusions

In this paper we have given some solutions and partial solutions to open problems in the book [19]. We briefly mention solutions to two other problems in [19].

Two texts $T$ and $T'$ are said to be *cousins* if

$\mathrm{content}(T) = \mathrm{content}(T')$ and

$(\exists m,n)(\forall i)[T(m+i) = T'(n+i)]$.

Thus $T$ and $T'$ are almost the same text for some language $L$. A machine is monotonic if it behaves the same way for all cousins, i.e., if it identifies $T$ (that is, $\mathbf{M}(T)\downarrow \wedge W_{\mathbf{M}(T)} = \mathrm{content}(T)$) then it identifies all cousins $T'$ of $T$. It can be shown that every $\mathcal{L} \in \mathbf{TxtEx}$ can be $\mathbf{TxtEx}$-identified by some monotonic machine. This solves an open question (open question 4.6.3A, page 92) in [19]. This result follows directly from Theorem 13 in [10].

Another problem in [19] dealt with *efficient identification.* A machine $\mathbf{M}$ converges on $T$ *faster than* $\mathbf{M}'$ iff the point of convergence for $\mathbf{M}$ on $T$ is earlier than that of $\mathbf{M}'$, i.e.,

$\max(\{n \mid \mathbf{M}(T[n]) \neq \mathbf{M}(T[n+1])\}) < \max(\{n \mid \mathbf{M}'(T[n]) \neq \mathbf{M}'(T[n+1])\})$.

A machine $\mathbf{M}$ *text efficiently* $\mathbf{TxtEx}$-*identifies* $\mathcal{L}$ iff

(a) $\mathbf{M}$ $\mathbf{TxtEx}$-identifies $\mathcal{L}$ and

(b) For all $F$, (may not be recursive) if $F$ $\mathbf{TxtEx}$-identifies $\mathcal{L}$ and $F$ converges on a text $T$ for $L \in \mathcal{L}$ faster than $\mathbf{M}$, then there exists a text $T'$ for $L' \in \mathcal{L}$ on which $\mathbf{M}$ converges faster than $F$.

It can be shown that there exists a class of svt languages which can be text-efficiently identified but is not recursively presentable [12], solving an open question (open question 8.2.4A, page 175) in [19].

## 8 Acknowledgements

We would like to thank John Case, Arun Sharma, Rajeev Raman and Lata Narayanan for helpful discussions.

## References

[1] D. Angluin and C. Smith. A survey of inductive inference: Theory and methods. *Computing Surveys*, 15:237–289, 1983.

[2] J. M. Barzdin. Two theorems on the limiting synthesis of functions. *In Theory of Algorithms and Programs, Latvian State University, Riga*, 210:82–88, 1974. In Russian.

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

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

[5] J. Case. The power of vacillation. In D. Haussler and L. Pitt, editors, *Proceedings of the Workshop on Computational Learning Theory*, pages 133–142. Morgan Kaufmann Publishers, Inc., 1988. Expanded in [8].

[6] J. Case and C. Lynes. Machine inductive inference and language identification. *Lecture Notes in Computer Science*, 140:107–115, 1982.

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

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

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

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

[11] M. A. Fulk and S. Jain. Learning in the presence of inaccurate information. In R. Rivest, D. Haussler, and M. K. Warmuth, editors, *Proceedings of the Second Annual Workshop on Computational Learning Theory, Santa Cruz, California*, pages 175–188. Morgan Kaufmann Publishers, Inc., August 1989.

[12] M. A. Fulk and S. Jain. Open problems in systems that learn. Technical Report 285, University of Rochester, 1989.

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

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

[15] S. Jain. *Learning in the Presence of Additional Information and Inaccurate Information*. PhD thesis, University of Rochester, 1990.

[16] S. Jain and M. Fulk. Open problems in systems that learn. In D Haussler and L. Pitt, editors, *Proceedings of the First Annual Workshop on Computational Learning Theory, MIT, Cambridge*, pages 425–426. Morgan Kaufmann Publishers, August 1988.

[17] R. Klette and R. Wiehagen. Research in the theory of inductive inference by GDR mathematicians – A survey. *Information Sciences*, 22:149–169, 1980.

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

[19] 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.

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

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

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

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