Synthesizing Noise-Tolerant Language Learners*

John Case
Department of CIS
University of Delaware
Newark, DE 19716, USA
Email: case@cis.udel.edu

Sanjay Jain
Department of ISCS
National University of Singapore
Singapore 119260
Email: sanjay@iscs.nus.edu.sg

Arun Sharma†
School of Computer Science and Engineering
The University of New South Wales
Sydney, NSW 2052, Australia
Email: arun@cse.unsw.edu.au

Abstract

An *index for an r.e. class of languages* (by definition) generates a sequence of grammars defining the class. An *index for an indexed family of languages* (by definition) generates a sequence of decision procedures defining the family.

F. Stephan's model of noisy data is employed, in which, roughly, correct data crops up infinitely often, and incorrect data only finitely often.

Studied, then, is the synthesis *from indices for r.e. classes and for indexed families of languages* of various kinds of *noise-tolerant* language-learners for the corresponding classes or families indexed.

Many positive results, as well as some negative results, are presented regarding the existence of such synthesizers. The proofs of most of the positive results yield, as pleasant corollaries, strict subset-principle or tell-tale style characterizations for the noise-tolerant learnability of the corresponding classes or families indexed.

1 Introduction

Ex-*learners*, when successful on an object input, (by definition) find a final correct program for that object after at most finitely many trial and error attempts [Gol67, BB75, CS83, CL82].[1]

For function learning, there is a learner-synthesizer algorithm **lsyn** so that, if **lsyn** is fed any procedure that lists programs for some (possibly infinite) class $\mathcal{S}$ of (total) functions, then **lsyn** outputs an **Ex**-learner successful on $\mathcal{S}$ [Gol67]. The learners so synthesized are called *enumeration techniques* [BB75, Ful90]. These enumeration techniques yield many positive learnability results, for example, that the class of all functions computable in time polynomial in the length

*This paper is dedicated to the memory of Mark Fulk.

†Research supported by Australian Research Council Grant A49530274.

[1] **Ex** is short for *explanatory*.

of input is **Ex**-learnable. The reader is referred to Jantke [Jan79] for a discussion of synthesizing learners for classes of recursive functions that are not necessarily recursively enumerable.

For language learning *from positive data and with learners outputting grammars*, [OSW88] provided an amazingly negative result: there is *no* learner-synthesizer algorithm **lsyn** so that, if **lsyn** is fed a *pair* of grammars g_1, g_2 for a language class $\mathcal{L} = \{L_1, L_2\}$, then **lsyn** outputs an **Ex**-learner successful, from positive data, on $\mathcal{L}$.[2] [BCJ96] showed how to circumvent *some* of the sting of this [OSW88] result by resorting to more general learners than **Ex**. Example more general learners are: **Bc**-*learners*, which, when successful on an object input, (by definition) find a final (possibly infinite) *sequence* of correct programs for that object after at most finitely many trial and error attempts [B$\bar{7}$4, CS83].[3] Of course, if suitable learner-synthesizer algorithm **lsyn** is fed procedures for listing *decision procedures* (instead of mere grammars), one also has more success at synthesizing learners. In fact the computational learning theory community has shown considerable interest (spanning at least from [Gol67] to [ZL95]) in language classes defined by r.e. listings of decision procedures. These classes are called *uniformly decidable* or *indexed families*. As is essentially pointed out in [Ang80], all of the formal language style example classes *are* indexed families. A sample result from [BCJ96] is: there *is* a learner-synthesizer algorithm **lsyn** so that, if **lsyn** is fed any procedure that lists decision procedures defining some indexed family $\mathcal{L}$ of languages *which can be* **Bc**-*learned from positive data with the learner outputting grammars*, then **lsyn** outputs a **Bc**-learner successful, from positive data, on $\mathcal{L}$. The proof of this positive result yielded the surprising characterization [BCJ96]: if there *is* an r.e. listing of decision procedures defining $\mathcal{L}$, i.e., if $\mathcal{L}$ is an indexed family, then: $\mathcal{L}$ can be **Bc**-learned from positive data with the learner outputting grammars iff

$$(\forall L \in \mathcal{L})(\exists S \subseteq L \mid S \text{ is finite})(\forall L' \in \mathcal{L} \mid S \subseteq L')[L' \not\subset L]. \tag{1}$$

(1) is Angluin's important Condition 2 from [Ang80], and it is referred to as the *subset principle*, in general a necessary condition for preventing overgeneralization in learning from positive data [Ang80, Ber85, ZLK95, KB92, Cas96]. As we will see, in the present paper, the proofs of most of our positive results which provide the *existence* of learner-synthesizers which synthesize *noise-tolerant* learners also yield pleasant characterizations which look like strict versions of the subset principle (1).[4]

We consider language learning from both texts (only positive data) and from informants (both positive and negative data), and we adopt Stephan's [Ste95, CJS96] noise model for the present study. Roughly, in this model correct information about an object occurs infinitely often while incorrect information occurs only finitely often. Hence, this model has the advantage that noisy data about an object nonetheless uniquely specifies that object. We note, though, that the presence of noise plays havoc with the learnability of many concrete classes that can be learned without noise. For example, the well-known class of *pattern languages* [Ang80] can be **Ex**-learned from texts but cannot be **Bc**-learned from noisy texts even if we allow the final grammars each to make finitely

[2] Again for language learning from positive data and with learners outputting grammars, a somewhat related negative result is provided by Kapur [Kap91]. He shows that one cannot algorithmically find an **Ex**-learning machine for **Ex**-*learnable* indexed families of recursive languages from an index of the class. This is a bit weaker than a closely related negative result from [BCJ96].

[3] **Bc** is short for *behaviorally correct*.

[4] For $\mathcal{L}$ either an indexed family or defined by some r.e. listing of grammars, the prior literature has many interesting characterizations of $\mathcal{L}$ being **Ex**-learnable from noise-free positive data, with and without extra restrictions. See, for example, [Ang80, Muk92, LZK96, dJK96]. For examples of characterization of learning from texts for not necessarily indexed families of languages see [JS94, JS97].

many mistakes. Fortunately, it *is* possible to **Ex**-learn the pattern languages *from informants* in the presence of noise, but a mind-change complexity price must be paid: any **Ex**-learner succeeding on the pattern languages from noisy informant must change its mind an unbounded finite number of times about the final grammar; however, some learner can succeed on the pattern languages from noise-*free* informants *and on its first guess as to a correct grammar* (see [LZK96]). The class of languages formed by taking the union of two pattern languages can be **Ex**-learned from texts [Shi83]; however, this class cannot be **Bc**-learned from noisy informants even if we allow the final grammars each to make finitely many mistakes.

In the present paper, we are concerned with learner-synthesizer algorithms which operate on procedures which list either grammars or decision procedures defining language classes and which output learners *which succeed in spite of receiving noisy data*.

We first consider, in Section 3, r.e. classes of r.e. languages, i.e., language classes defined by an r.e. listing of grammars. For this case we show that the synthesis of *noise-tolerant* learners is possible, *only* for **Bc**-learners, operating on texts or informants, *whose final grammars each make finitely many mistakes*. In the process we also characterize, *for r.e. classes of r.e. languages*, the power of these kinds of noise-tolerant **Bc**-learners by principles similar to the subset principle in (1) above.

For indexed families of languages, the picture is a lot more encouraging. We show, in Section 4, for indexed families of languages, the surprising facts that synthesis *can* be achieved for a variety of *noise-tolerant* learners: 1. for **Bc**-learners, operating from either texts or informants, whose final grammars are allowed to make either a bounded or an unbounded finite number of mistakes;[5] 2. for **Ex**-learners, operating from *texts*, whose final grammar is allowed to make an *un*bounded finite number of mistakes; and 3. for **Ex**-learners, operating from *informants*, whose final grammar is allowed to make an *bounded* finite number of mistakes. In each of these cases there is a corresponding pleasant characterization *for indexed families of languages*, of the power of these kinds of noise-tolerant learners by strict subset principles similar to (1) above. Here is an example. If $\mathcal{L}$ is an indexed family, then: $\mathcal{L}$ can be noise-tolerantly **Bc**-learned from positive data with the learner outputting grammars iff

$$(\forall L, L' \in \mathcal{L})[L \subseteq L' \Rightarrow L = L']. \tag{2}$$

We also show that, for indexed families of languages, while synthesis of noise-tolerant learners is *not* possible for **Ex**-learners, operating from *informants*, whose final grammar is allowed to make an *un*bounded finite number of mistakes; it *is* possible for **Ex**-learners, operating from *texts*, whose final grammar is allowed to make $\leq n$ mistakes, but where the noise-tolerant synthesized learner may double the number of mistakes up to $2n$. If $n = 0$, then we get a characterization of indexed families $\mathcal{L}$ noise-tolerantly **Ex**-learnable, from texts, also by (2) above!

2 Preliminaries

2.1 Notation and identification criteria

The recursion theoretic notions are from the books of Odifreddi [Odi89] and Soare [Soa87]. $N = \{0, 1, 2, \ldots\}$ is the set of all natural numbers, and this paper considers r.e. subsets L of N. $N^+ = \{1, 2, 3, \ldots\}$, the set of all positive integers. All conventions regarding range of variables apply,

[5] This includes the bound of 0: *no* mistakes!

with or without decorations[6], unless otherwise specified. We let c, e, i, j, k, l, m, n, q, s, t, u, v, w, x, y, z, range over N. $\emptyset, \in, \subseteq, \supseteq, \subset, \supset$ denote empty set, member of, subset, superset, proper subset, and proper superset respectively. $\max(), \min(), \mathrm{card}()$ denote the maximum, minimum, and cardinality of a set respectively, where by convention $\max(\emptyset) = 0$ and $\min(\emptyset) = \infty$. $\mathrm{card}(S) \leq *$ means cardinality of set S is finite. a, b range over $N \cup \{*\}$. $\langle \cdot, \cdot \rangle$ stands for an arbitrary but fixed, one to one, computable encoding of all pairs of natural numbers onto N. π_1 and π_2 denote the corresponding projection functions; that is, $\pi_1(\langle x, y \rangle) = x$ and $\pi_2(\langle x, y \rangle) = y$. $\langle \cdot, \cdot, \cdot \rangle$, similarly denotes a computable, 1–1 encoding of all triples of natural numbers onto N. $\overline{L}$ denotes the complement of set L. χ_L denotes the characteristic function of set L. $L_1 \mathbf{\Delta} L_2$ denotes the symmetric difference of L_1 and L_2, i.e., $L_1 \mathbf{\Delta} L_2 = (L_1 - L_2) \cup (L_2 - L_1)$. $L_1 =^a L_2$ means that $\mathrm{card}(L_1 \mathbf{\Delta} L_2) \leq a$. Quantifiers $\forall^\infty, \exists^\infty$, and $\exists!$ denote for all but finitely many, there exist infinitely many, and there exists a unique respectively.

$\mathcal{R}$ denotes the set of total computable functions (recursive functions) from N to N. f, g, range over total computable functions. $\mathcal{E}$ denotes the set of all recursively enumerable sets. L, ranges over $\mathcal{E}$. $\mathcal{L}$, ranges over subsets of $\mathcal{E}$. φ denotes a standard acceptable programming system (acceptable numbering). φ_i denotes the function computed by the i-th program in the programming system φ. We also call i a program or index for φ_i. For a (partial) function η, domain(η) and range(η) respectively denote the domain and range of partial function η. We often write $\eta(x)\downarrow$ ($\eta(x)\uparrow$) to denote that $\eta(x)$ is defined (undefined). W_i denotes the domain of φ_i. W_i is considered as the language enumerated by the i-th program in φ system, and we say that i is a grammar or index for W_i. Φ denotes a standard Blum complexity measure [Blu67] for the programming system φ. $W_{i,s} = \{x < s \mid \Phi_i(x) < s\}$.

A *text* is a mapping from N to $N \cup \{\#\}$. We let T, range over texts. content(T) is defined to be the set of natural numbers in the range of T (i.e. $\mathrm{content}(T) = \mathrm{range}(T) - \{\#\}$). T is a *text for* L iff $\mathrm{content}(T) = L$. That means a text for L is an infinite sequence whose range, except for a possible $\#$, is just L.

An *information sequence or informant* is a mapping from N to $(N \times N) \cup \{\#\}$. We let I, range over informants. content(I) is defined to be the set of pairs in the range of I (i.e., $\mathrm{content}(I) = \mathrm{range}(I) - \{\#\}$). An *informant for* L is an infinite sequence I such that $\mathrm{content}(I) = \{(x, b) \mid \chi_L(x) = b\}$. It is useful to consider canonical information sequence for L. I is a canonical information sequence for L iff $I(x) = (x, \chi_L(x))$. We sometimes abuse notation and refer to the canonical information sequence for L by χ_L.

σ and τ, range over finite initial segments of texts or information sequences, where the context determines which is meant. We denote the set of finite initial segments of texts by SEG and set of finite initial segments of information sequences by SEQ. We use $\sigma \preceq T$ (respectively, $\sigma \preceq I$, $\sigma \preceq \tau$) to denote that σ is an initial segment of T (respectively, I, τ). $|\sigma|$ denotes the length of σ. $T[n]$ denotes the initial segment of T of length n. Similarly, $I[n]$ denotes the initial segment of I of length n. Let $T[m:n]$ denote the segment $T(m), T(m+1), \ldots, T(n-1)$ (i.e. $T[n]$ with the first m elements, $T[m]$, removed). $I[m:n]$ is defined similarly. $\sigma \diamond \tau$ (respectively, $\sigma \diamond T$, $\sigma \diamond I$) denotes the concatenation of σ and τ (respectively, concatenation of σ and T, concatenation of σ and I). We sometimes abuse notation and say $\sigma \diamond w$ to denote the concatenation of σ with the sequence of one element w.

A *learning machine* $\mathbf{M}$ is a mapping from initial segments of texts (information sequences) to N. We say that $\mathbf{M}$ converges on T to i, (written: $\mathbf{M}(T)\downarrow = i$) iff, for all but finitely many n,

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

$\mathbf{M}(T[n]) = i$. Convergence on information sequences is defined similarly.

Let $\mathrm{ProgSet}(\mathbf{M}, \sigma) = \{\mathbf{M}(\tau) \mid \tau \subseteq \sigma\}$.

Definition 1 Suppose $a \in N \cup \{*\}$.
(a) Below, for each of several learning criteria $\mathbf{J}$, we define what it means for a machine $\mathbf{M}$ to $\mathbf{J}$-*identify* a language L from a text T or informant I.

- [Gol67, CL82, BB75] $\mathbf{M}$ $\mathbf{TxtEx}^a$-*identifies* L *from text* T iff $(\exists i \mid W_i =^a L)[\mathbf{M}(T)\downarrow = i]$.

- [Gol67, CL82, BB75] $\mathbf{M}$ $\mathbf{InfEx}^a$-*identifies* L *from informant* I iff $(\exists i \mid W_i =^a L)[\mathbf{M}(I)\downarrow = i]$.

- [B$\bar{7}$4, CL82]. $\mathbf{M}$ $\mathbf{TxtBc}^a$-*identifies* L *from text* T iff $(\forall^\infty n)[W_{\mathbf{M}(T[n])} =^a L]$.

  $\mathbf{InfBc}^a$-identification is defined similarly.

- [Cas96, BP73]. $\mathbf{M}$ $\mathbf{TxtFex}^a$-*identifies* L *from text* T iff $(\exists S \mid 0 < \mathrm{card}(S) < \infty \ \wedge \ (\forall i \in S)[W_i =^a L])(\forall^\infty n)[\mathbf{M}(T[n]) \in S]$.

  $\mathbf{InfFex}^a$ is defined similarly.

  Based on the definition of $\mathbf{TxtFex}^a$ and $\mathbf{InfFex}^a$-identification, we sometimes also say that $\mathbf{M}$ on T converges to a finite set S of grammars iff $(\forall^\infty n)\,[\mathbf{M}(T[n]) \in S]$. If no such S exists, then we say that $\mathbf{M}$ on T does not converge to a finite set of grammars. Similarly we define convergence and divergence on information sequences.

  $\mathbf{Last}_*(\mathbf{M}, \sigma)$ denotes the set of grammars output by $\mathbf{M}$ on σ. That is, $\mathbf{Last}_*(\mathbf{M}, \sigma) = \{\mathbf{M}(\tau) \mid \tau \preceq \sigma\}$.

  If $\lim_{n\to\infty} \mathbf{Last}_*(\mathbf{M}, T[n])\downarrow$, then we say that $\mathbf{Last}_*(\mathbf{M}, T) = \lim_{n\to\infty} \mathbf{Last}_*(\mathbf{M}, T[n])$. Otherwise $\mathbf{Last}_*(\mathbf{M}, T)$ is undefined. $\mathbf{Last}_*(\mathbf{M}, I)$ is defined similarly.

(b) Suppose $\mathbf{J} \in \{\mathbf{TxtEx}^a, \mathbf{TxtFex}^a, \mathbf{TxtBc}^a\}$.

$\mathbf{M}$ $\mathbf{J}$-*identifies* L iff, for all texts T for L, $\mathbf{M}$ $\mathbf{J}$-identifies L from T. In this case we also write $L \in \mathbf{J}(\mathbf{M})$.

We say that $\mathbf{M}$ $\mathbf{J}$-identifies $\mathcal{L}$ iff $\mathbf{M}$ $\mathbf{J}$-identifies each $L \in \mathcal{L}$.

$\mathbf{J} = \{\mathcal{L} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{J}(\mathbf{M})]\}$.

(c) Suppose $\mathbf{J} \in \{\mathbf{InfEx}^a, \mathbf{InfFex}^a, \mathbf{InfBc}^a\}$.

$\mathbf{M}$ $\mathbf{J}$-*identifies* L iff, for all information sequences I for L, $\mathbf{M}$ $\mathbf{J}$-identifies L from I. In this case we also write $L \in \mathbf{J}(\mathbf{M})$.

We say that $\mathbf{M}$ $\mathbf{J}$-identifies $\mathcal{L}$ iff $\mathbf{M}$ $\mathbf{J}$-identifies each $L \in \mathcal{L}$.

$\mathbf{J} = \{\mathcal{L} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{J}(\mathbf{M})]\}$.

We often write $\mathbf{TxtEx}^0$ as $\mathbf{TxtEx}$, $\mathbf{TxtBc}$ for $\mathbf{TxtBc}^0$ and $\mathbf{TxtFex}^0$ as $\mathbf{TxtFex}$. Similar convention applies to other criteria of inference considered in this paper.

Several proofs in this paper depend on the concept of locking sequence.

Definition 2 (Based on [BB75]) Suppose $a \in N \cup \{*\}$.

(a) σ is said to be a $\mathbf{TxtEx}^a$-*locking sequence for* $\mathbf{M}$ *on* L iff, $\mathrm{content}(\sigma) \subseteq L$, $W_{\mathbf{M}(\sigma)} =^a L$, and $(\forall \tau \mid \mathrm{content}(\tau) \subseteq L)[\mathbf{M}(\sigma \diamond \tau) = \mathbf{M}(\sigma)]$.

(b) σ is said to be a $\mathbf{TxtBc}^a$-*locking sequence for* $\mathbf{M}$ *on* L iff, $\mathrm{content}(\sigma) \subseteq L$, and $(\forall \tau \mid \mathrm{content}(\tau) \subseteq L)[W_{\mathbf{M}(\sigma\diamond\tau)} =^a L]$.

(c) σ is said to be a $\mathbf{TxtFex}^a$-*locking sequence for* $\mathbf{M}$ *on* L iff, $\mathrm{content}(\sigma) \subseteq L$, and there exists a set S such that

(c.1) $\mathrm{card}(S) < \infty$,
(c.2) $S \subseteq \mathbf{Last}_*(\mathbf{M}, \sigma)$,
(c.3) $(\forall i \in S)[W_i =^a L]$, and
(c.4) $(\forall \tau \mid \mathrm{content}(\tau) \subseteq L)[\mathbf{M}(\sigma \diamond \tau) \in S]$.

Lemma 1 (Based on [BB75]) *Suppose $a \in N \cup \{*\}$. Suppose $\mathbf{J} \in \{\mathbf{TxtEx}^a, \mathbf{TxtFex}^a, \mathbf{TxtBc}^a\}$. If $\mathbf{M}$ $\mathbf{J}$-identifies L then there exists a $\mathbf{J}$-locking sequence for $\mathbf{M}$ on L.*

Next we prepare to introduce our noisy inference criteria, and, in that interest, we define some ways to calculate the number of occurrences of words in (initial segments of) a text or informant. For $\sigma \in$ SEG, and text T, let

$$\mathrm{occur}(\sigma, w) \stackrel{\mathrm{def}}{=} \mathrm{card}(\{j \mid j < |\sigma| \ \wedge \ \sigma(j) = w\}) \text{ and}$$
$$\mathrm{occur}(T, w) \stackrel{\mathrm{def}}{=} \mathrm{card}(\{j \mid j \in N \ \wedge \ T(j) = w\}).$$

For $\sigma \in$ SEQ and information sequence I, $\mathrm{occur}(\cdot, \cdot)$ is defined similarly except that w is replaced by (v, b).

For any language L, $\mathrm{occur}(T, L) \stackrel{\mathrm{def}}{=} \Sigma_{x \in L}\, \mathrm{occur}(T, x)$. It is useful to introduce the set of positive and negative occurrences in (initial segment of) an informant. Suppose $\sigma \in$ SEQ

$$\begin{aligned}
\mathrm{PosInfo}(\sigma) &\stackrel{\mathrm{def}}{=} \{v \mid \mathrm{occur}(\sigma, (v,1)) \geq \mathrm{occur}(\sigma, (v,0)) \wedge \mathrm{occur}(\sigma, (v,1)) \geq 1\} \\
\mathrm{NegInfo}(\sigma) &\stackrel{\mathrm{def}}{=} \{v \mid \mathrm{occur}(\sigma, (v,1)) < \mathrm{occur}(\sigma, (v,0)) \wedge \mathrm{occur}(\sigma, (v,0)) \geq 1\}
\end{aligned}$$

That means, that $\mathrm{PosInfo}(\sigma) \cup \mathrm{NegInfo}(\sigma)$ is just the set of all v such that either $(v, 0)$ or $(v, 1)$ occurs in σ. Then $v \in \mathrm{PosInfo}(\sigma)$ if $(v, 1)$ occurs at least as often as $(v, 0)$ and $v \in \mathrm{NegInfo}(\sigma)$ otherwise.

Similarly,

$$\begin{aligned}
\mathrm{PosInfo}(I) &= \{v \mid \mathrm{occur}(I, (v,1)) \geq \mathrm{occur}(I, (v,0)) \wedge \mathrm{occur}(I, (v,1)) \geq 1\} \\
\mathrm{NegInfo}(I) &= \{v \mid \mathrm{occur}(I, (v,1)) < \mathrm{occur}(I, (v,0)) \wedge \mathrm{occur}(I, (v,0)) \geq 1\}
\end{aligned}$$

where, if $\mathrm{occur}(I, (v, 0)) = \mathrm{occur}(I, (v, 1)) = \infty$, then we place v in $\mathrm{PosInfo}(I)$ (this is just to make the definition precise; we will not need this for criteria of inference discussed in this paper).

Definition 3 [Ste95] An information sequence I is a *noisy information sequence* (or noisy informant) for L iff $(\forall x)\,[\mathrm{occur}(I, (x, \chi_L(x))) = \infty \ \wedge \ \mathrm{occur}(I, (x, \chi_{\overline{L}}(x))) < \infty]$. A text T is a *noisy text* for L iff $(\forall x \in L)[\mathrm{occur}(T, x) = \infty]$ and $\mathrm{occur}(T, \overline{L}) < \infty$.

On one hand, both concepts are similar since $L = \{x \mid \mathrm{occur}(I, (x, 1)) = \infty\} = \{x \mid \mathrm{occur}(T, x) = \infty\}$. On the other hand, the concepts differ in the way they treat errors. In the case of informant every false item $(x, \chi_{\overline{L}}(x))$ may occur a finite number of times. In the case of text, it is mathematically more interesting to require, as we do, that the *total* amount of false information has to be finite.[7]

[7] The alternative of allowing *each* false item in a text to occur finitely often is too restrictive; it would, then, be impossible to learn even the class of all singleton sets [Ste95].

Definition 4 [Ste95, CJS96] *Suppose $a \in N \cup \{*\}$. Suppose $\mathbf{J} \in \{\mathbf{TxtEx}^a, \mathbf{TxtFex}^a, \mathbf{TxtBc}^a\}$. Then $\mathbf{M}$ $\mathbf{NoisyJ}$-identifies L iff, for all noisy texts T for L, $\mathbf{M}$ $\mathbf{J}$-identifies L from T. In this case we write $L \in \mathbf{NoisyJ}(\mathbf{M})$.*

$\mathbf{M}$ $\mathbf{NoisyJ}$-identifies a class $\mathcal{L}$ iff $\mathbf{M}$ $\mathbf{NoisyJ}$-identifies each $L \in \mathcal{L}$.

$\mathbf{NoisyJ} = \{\mathcal{L} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{NoisyJ}(\mathbf{M})]\}$.

Inference criteria for learning from noisy informants are defined similarly.

Several proofs use the existence of locking sequences. Definition of locking sequences for learning from noisy texts is similar to that of learning from noise free texts (we just drop the requirement that content$(\sigma) \subseteq L$). However, definition of locking sequence for learning from noisy informant is more involved.

Definition 5 [CJS96] Suppose $a \in N \cup \{*\}$.

(a) σ is said to be a $\mathbf{NoisyTxtEx}^a$-*locking sequence for* $\mathbf{M}$ *on* L iff, $W_{\mathbf{M}(\sigma)} =^a L$, and $(\forall \tau \mid \text{content}(\tau) \subseteq L)[\mathbf{M}(\sigma \diamond \tau) = \mathbf{M}(\sigma)]$.

(b) σ is said to be a $\mathbf{NoisyTxtBc}^a$-*locking sequence for* $\mathbf{M}$ *on* L iff $(\forall \tau \mid \text{content}(\tau) \subseteq L)[W_{\mathbf{M}(\sigma\diamond\tau)} =^a L]$.

(c) σ is said to be a $\mathbf{NoisyTxtFex}^a$-*locking sequence for* $\mathbf{M}$ *on* L iff there exists a set S such that

(c.1) $\text{card}(S) < \infty$,
(c.2) $S \subseteq \mathbf{Last}_*(\mathbf{M}, \sigma)$,
(c.3) $(\forall i \in S)[W_i =^a L]$, and
(c.4) $(\forall \tau \mid \text{content}(\tau) \subseteq L)[\mathbf{M}(\sigma \diamond \tau) \in S]$.

For defining locking sequences for learning from noisy informant, we need the following.

Definition 6 $\text{Inf}[S, L] \stackrel{\text{def}}{=} \{\tau \mid (\forall x \in S)\,[\text{occur}(\tau, (x, \chi_{\overline{L}}(x))) = 0]\}$.

Definition 7 Suppose $a \in N \cup \{*\}$.

(a) σ is said to be a $\mathbf{NoisyInfEx}^a$-*locking sequence for* $\mathbf{M}$ *on* L iff, $\text{PosInfo}(\sigma) \subseteq L$, $\text{NegInfo}(\sigma) \subseteq \overline{L}$, $W_{\mathbf{M}(\sigma)} =^a L$, and $(\forall \tau \in \text{Inf}[\text{PosInfo}(\sigma) \cup \text{NegInfo}(\sigma), L])[\mathbf{M}(\sigma \diamond \tau) = \mathbf{M}(\sigma)]$.

(b) σ is said to be a $\mathbf{NoisyInfBc}^a$-*locking sequence for* $\mathbf{M}$ *on* L iff, $\text{PosInfo}(\sigma) \subseteq L$, $\text{NegInfo}(\sigma) \subseteq \overline{L}$, and $(\forall \tau \in \text{Inf}[\text{PosInfo}(\sigma) \cup \text{NegInfo}(\sigma), L])[W_{\mathbf{M}(\sigma\diamond\tau)} =^a L]$.

(c) σ is said to be a $\mathbf{NoisyInfFex}^a$-*locking sequence for* $\mathbf{M}$ *on* L iff, $\text{PosInfo}(\sigma) \subseteq L$, $\text{NegInfo}(\sigma) \subseteq \overline{L}$, and there exists a set S such that

(c.1) $\text{card}(S) < \infty$,
(c.2) $S \subseteq \mathbf{Last}_*(\mathbf{M}, \sigma)$,
(c.3) $(\forall i \in S)[W_i =^a L]$, and
(c.4) $(\forall \tau \in \text{Inf}[\text{PosInfo}(\sigma) \cup \text{NegInfo}(\sigma), L])[\mathbf{M}(\sigma \diamond \tau) \in S]$.

For the criteria of noisy inference discussed in this paper one can prove the existence of a locking sequence as was done in [Ste95, Theorem 2, proof for $\mathbf{NoisyEx} \subseteq \mathbf{Ex}_0[K]$].

Proposition 1 *[CJS96] Suppose $a \in N \cup \{*\}$.*

If $\mathbf{M}$ learns L from noisy text or informant according to one of the criteria $\mathbf{NoisyTxtEx}^a$, $\mathbf{NoisyTxtFex}^a$, and $\mathbf{NoisyTxtBc}^a$, $\mathbf{NoisyInfEx}^a$, $\mathbf{NoisyInfFex}^a$, and $\mathbf{NoisyInfBc}^a$, then there exists a corresponding locking sequence for $\mathbf{M}$ on L.

2.2 Recursively enumerable classes and indexed families

The aim of this paper is to consider (effective) learnability of enumerable classes and indexed families of recursive languages. To this end we define, for all i, $\mathcal{C}_i = \{W_j \mid j \in W_i\}$. For a decision procedure j, let $U_j = \{x \mid \varphi_j(x) = 1\}$. For a decision procedure j, we let $U_j[n]$ denote $\{x \in U_j \mid x < n\}$.

For all i,

$$\mathcal{U}_i = \begin{cases} \{U_j \mid j \in W_i\}, & \text{if } (\forall j \in W_i)[j \text{ is a decision procedure}]; \\ \emptyset, & \text{otherwise.} \end{cases}$$

2.3 Some previous results on noisy text/informant identification

We first state some results from [CJS96] which are useful. We let $2* \stackrel{\text{def}}{=} *$.

Theorem 1 [CJS96] *Suppose $a \in N \cup \{*\}$. $\mathcal{L} \in \mathbf{NoisyTxtBc}^a \Rightarrow [(\forall L \in \mathcal{L})(\forall L' \in \mathcal{L} \mid L' \subseteq L)[L =^{2a} L']]$.*

As an immediate corollary to Proposition 1 we have the following two theorems,

Theorem 2 *Suppose $a \in N \cup \{*\}$. Suppose $\mathcal{L} \in \mathbf{NoisyInfBc}^a$. Then for all $L \in \mathcal{L}$, there exists an n such that $(\forall L' \in \mathcal{L} \mid \{x \in L \mid x \leq n\} = \{x \in L' \mid x \leq n\})[L =^{2a} L']$.*

Theorem 3 *Suppose $a \in N \cup \{*\}$. Suppose $\mathcal{L} \in \mathbf{NoisyInfEx}^a$. Then for all $L \in \mathcal{L}$, there exist n and S such that $(\forall L' \in \mathcal{L} \mid \{x \in L \mid x \leq n\} = \{x \in L' \mid x \leq n\})[(L \mathbf{\Delta} S) =^a L']$.*

As a corollary to Theorem 3 we have

Theorem 4 *Suppose $a \in N \cup \{*\}$. Suppose $\mathcal{L} \in \mathbf{NoisyInfEx}^a$. Then for all $L \in \mathcal{L}$, there exists an n such that $(\forall L' \in \mathcal{L} \mid \{x \in L \mid x \leq n\} = \{x \in L' \mid x \leq n\})[L =^a L']$.*

Similarly, one can show,

Theorem 5 *Suppose $\mathcal{L} \in \mathbf{NoisyInfFex}^a$. Then for all $L \in \mathcal{L}$, there exists an n such that $(\forall L' \in \mathcal{L} \mid \{x \in L \mid x \leq n\} = \{x \in L' \mid x \leq n\})[L =^a L']$.*

The following theorem was proved in [CJS96].

Theorem 6 *Suppose $a \in N \cup \{*\}$. Then $\mathbf{NoisyInfBc}^a \cup \mathbf{NoisyTxtBc}^a \subseteq \mathbf{TxtBc}^a$ and $\mathbf{NoisyInfEx}^a \cup \mathbf{NoisyTxtEx}^a \subseteq \mathbf{TxtEx}^a$.*

The following proposition is easy to prove:

Proposition 2 *Suppose $\mathcal{L}$ is a finite class of languages such that for all $L, L' \in \mathcal{L}$, $L \subseteq L' \Rightarrow L = L'$. Then, $\mathcal{L} \in \mathbf{NoisyTxtEx} \cap \mathbf{NoisyInfEx}$.*

Suppose $\mathcal{L}$ is a finite class of languages. Then, $\mathcal{L} \in \mathbf{NoisyInfEx}$.

3 Identification from enumeration procedures

In this section we show that effective synthesis from enumeration procedures for noisy inference criteria can be done only in the case of $\mathbf{NoisyTxtBc}^*$ and $\mathbf{NoisyInfBc}^*$-identification criteria. We also characterize $\mathbf{NoisyTxtBc}^*$ and $\mathbf{NoisyInfBc}^*$ in the process. We first consider cases in which effective synthesis is not possible.

3.1 When effective synthesis is not possible

As a corollary to Theorem 7 below we immediately have that effective synthesis of learning machines is not possible for the following noisy inference criteria.

- $\mathbf{NoisyTxtBc}^n$, for $n \in N$;
- $\mathbf{NoisyTxtEx}^a$, for $a \in N \cup \{*\}$;
- $\mathbf{NoisyInfBc}^n$, for $n \in N$; and
- $\mathbf{NoisyInfEx}^a$, for $a \in N \cup \{*\}$.

Theorem 7 *NOT* $(\exists f \in \mathcal{R})(\exists n \in N)(\forall x \mid \mathcal{C}_x \in \mathbf{NoisyTxtEx} \cap \mathbf{NoisyInfEx})[\mathcal{C}_x \subseteq \mathbf{TxtBc}^n(\mathbf{M}_{f(x)}) \cup \mathbf{TxtFex}^*(\mathbf{M}_{f(x)})]$.

The above theorem follows as a corollary to Proposition 3 and Theorem 8 below.
Remark: The reader should note that Theorem 7 and other negative results of this paper hold even when "$(\exists f \in \mathcal{R})$" is replaced by "($\exists$ a limiting computable f)". Intuitively, f is *limiting-computable* $\stackrel{\text{def}}{\Leftrightarrow}$ there exists a computable function $g : N^2 \to N$ such that for each x, $f(x) = \lim_{s\to\infty} g(x, s)$. In this case we say that f is limiting computable as witnessed by g. Theorem 7 and the other negative results in this paper all hold in this stronger sense. In other words, not only, as they say, is there no *computable* function f such that ..., but, for these results, there is no *limiting-computable* function f such that ... . This follows from the fact that, a positive synthesis result (for learning criteria **Ex** or **Bc**, with text or informant, with or without noise) using a limiting computable function to generate the learner implies a corresponding positive synthesis result using a computable function to generate the learner. To see this, for any limiting computable function f (as witnessed by g), let f' be a computable function such that $\mathbf{M}_{f'(i)}(\sigma) = \mathbf{M}_{g(i,|\sigma|)}(\sigma)$. It is easy to note that $\mathbf{M}_{f'(i)}$ identifies the class $\mathcal{C}_i$ ($\mathcal{U}_i$) under criteria $\mathbf{TxtEx}^a$, $\mathbf{TxtBc}^a$, $\mathbf{InfEx}^a$, $\mathbf{InfBc}^a$, with or without noise, iff $\mathbf{M}_{\lim_{n\to\infty} g(i,n)} = M_{f(i)}$ correspondingly identifies $\mathcal{C}_i$ ($\mathcal{U}_i$).

Proposition 3 *For a language* L, *let* $\mathrm{Cyl}(L) = \{\langle x, y\rangle \mid x \in L \ \wedge \ y \in N\}$. *For a class* $\mathcal{L}$, *let* $\mathrm{Cyl}(\mathcal{L}) = \{\mathrm{Cyl}(L) \mid L \in \mathcal{L}\}$.
(a) $(\forall \mathcal{L})[\mathcal{L} \in \mathbf{NoisyTxtEx}$ *iff* $\mathrm{Cyl}(\mathcal{L}) \in \mathbf{NoisyTxtEx}]$.
(b) $(\forall \mathcal{L})[\mathcal{L} \in \mathbf{NoisyInfEx}$ *iff* $\mathrm{Cyl}(\mathcal{L}) \in \mathbf{NoisyInfEx}]$.
(c) $(\exists f \in \mathcal{R})(\forall \mathcal{L})[\mathrm{Cyl}(\mathcal{L}) \in \mathbf{TxtBc}^n(\mathbf{M}_i) \Rightarrow \mathrm{Cyl}(\mathcal{L}) \in \mathbf{TxtBc}(\mathbf{M}_{f(i)})]$.
(d) $(\exists f \in \mathcal{R})(\forall \mathcal{L})[\mathrm{Cyl}(\mathcal{L}) \in \mathbf{TxtFex}^*(\mathbf{M}_i) \Rightarrow \mathrm{Cyl}(\mathcal{L}) \in \mathbf{TxtBc}(\mathbf{M}_{f(i)})]$.
(e) $(\exists f \in \mathcal{R})(\forall \mathcal{L})[\mathcal{L} \in \mathbf{TxtBc}(\mathbf{M}_i) \Rightarrow \mathrm{Cyl}(\mathcal{L}) \in \mathbf{TxtBc}(\mathbf{M}_{f(i)})]$.
(f) $(\exists f \in \mathcal{R})(\forall \mathcal{L})[\mathrm{Cyl}(\mathcal{L}) \in \mathbf{TxtBc}(\mathbf{M}_i) \Rightarrow \mathcal{L} \in \mathbf{TxtBc}(\mathbf{M}_{f(i)})]$.

PROOF. Let g be a recursive function such that for a grammar i, $W_{g(i)} = \{\langle x, y\rangle \mid x \in W_i \wedge y \in N\}$.

Let h_m be a recursive function such that, for a grammar i, $W_{h_m(i)} = \{x \mid \mathrm{card}(\{y \mid \langle x, y\rangle \in W_i\}) \geq m\}$.

(a) ($\Rightarrow$) Suppose $\mathcal{L} \subseteq \mathbf{NoisyTxtEx}(\mathbf{M})$. Let F be a recursive function from SEG to SEG such that $|F(\sigma)| = |\sigma|$ and for $m < |\sigma|$,

$$F(\sigma)(m) = \begin{cases} x, & \text{if } \sigma(m) = \langle x, 0\rangle; \\ \#, & \text{if for all } x,\ \sigma(m) \neq \langle x, 0\rangle. \end{cases}$$

It is easy to verify that if T is a noisy text for $\mathrm{Cyl}(L)$, then $\bigcup_{m\in N} F(T[m])$ is a noisy text for L.

Let $\mathbf{M}'$ be defined as follows: $\mathbf{M}'(\sigma) = g(\mathbf{M}(F(\sigma)))$. It is easy to verify that if $\mathbf{M}$ **NoisyTxtEx**-identifies L then $\mathbf{M}'$ **NoisyTxtEx**-identifies $\mathrm{Cyl}(L)$. Thus, $\mathrm{Cyl}(\mathcal{L}) \in$ **NoisyTxtEx**.

($\Leftarrow$) Suppose $\mathrm{Cyl}(\mathcal{L}) \subseteq \mathbf{NoisyTxtEx}(\mathbf{M})$. Let F be a recursive function from SEG to SEG such that (i) for all $\sigma \subseteq \tau$, $F(\sigma) \subseteq F(\tau)$, and (ii) for all x, y,

$$\mathrm{occur}(F(\sigma), \langle x, y\rangle) = \begin{cases} \mathrm{occur}(\sigma, x), & \text{if } \mathrm{occur}(\sigma, x) \geq y; \\ 0, & \text{otherwise.} \end{cases}$$

It is easy to verify that if T is a noisy text for L, then $\bigcup_{m\in N} F(T[m])$ is a noisy text for $\mathrm{Cyl}(L)$.

Let $\mathbf{M}'$ be defined as follows: $\mathbf{M}'(\sigma) = h_1(\mathbf{M}(F(\sigma)))$. It is easy to verify that if $\mathbf{M}$ **NoisyTxtEx**-identifies $\mathrm{Cyl}(L)$ then $\mathbf{M}'$ **NoisyTxtEx**-identifies L. Thus, $\mathcal{L} \in$ **NoisyTxtEx**.

(b) ($\Rightarrow$) Suppose $\mathcal{L} \subseteq \mathbf{NoisyInfEx}(\mathbf{M})$. Let F be a recursive function from SEQ to SEQ such that $|F(\sigma)| = |\sigma|$ and for $m < |\sigma|$,

$$F(\sigma)(m) = \begin{cases} (x, b), & \text{if } \sigma(m) = (\langle x, 0\rangle, b); \\ \#, & \text{if for all } x \text{ and } b,\ \sigma(m) \neq (\langle x, 0\rangle, b). \end{cases}$$

It is easy to verify that if I is a noisy informant for $\mathrm{Cyl}(L)$, then $\bigcup_{m\in N} F(I[m])$ is a noisy informant for L.

Let $\mathbf{M}'$ be defined as follows: $\mathbf{M}'(\sigma) = g(\mathbf{M}(F(\sigma)))$. It is easy to verify that if $\mathbf{M}$ **NoisyInfEx**-identifies L then $\mathbf{M}'$ **NoisyInfEx**-identifies $\mathrm{Cyl}(L)$. Thus, $\mathrm{Cyl}(\mathcal{L}) \in$ **NoisyInfEx**.

($\Leftarrow$) Suppose $\mathrm{Cyl}(\mathcal{L}) \subseteq \mathbf{NoisyInfEx}(\mathbf{M})$. Let F be a recursive function from SEQ to SEQ such that (i) for all $\sigma \subseteq \tau$, $F(\sigma) \subseteq F(\tau)$, and (ii) for all x, y, b,

$$\mathrm{occur}(F(\sigma), (\langle x, y\rangle, b)) = \begin{cases} \mathrm{occur}(\sigma, (x, b)), & \text{if } \mathrm{occur}(\sigma, (x, b)) \geq y; \\ 0, & \text{otherwise.} \end{cases}$$

It is easy to verify that if I is a noisy informant for L, then $\bigcup_{m\in N} F(I[m])$ is a noisy informant for $\mathrm{Cyl}(L)$.

Let $\mathbf{M}'$ be defined as follows: $\mathbf{M}'(\sigma) = h_1(\mathbf{M}(F(\sigma)))$. It is easy to verify that if $\mathbf{M}$ **NoisyInfEx**-identifies $\mathrm{Cyl}(L)$ then $\mathbf{M}'$ **NoisyInfEx**-identifies L. Thus, $\mathcal{L} \in$ **NoisyInfEx**.

(c) Suppose f is a recursive function such that, for all i and σ, $\mathbf{M}_{f(i)}(\sigma) = h_{2n+1}(\mathbf{M}_i(\sigma))$. It is easy to verify that, for any L, if $\mathrm{Cyl}(L) \in \mathbf{TxtBc}^n(\mathbf{M}_i)$, then $\mathrm{Cyl}(L) \in \mathbf{TxtBc}(\mathbf{M}_{f(i)})$. Part (c) follows.

(d) Suppose f is a recursive function such that, for all i and σ, $\mathbf{M}_{f(i)}(\sigma) = h_{|\sigma|}(\mathbf{M}_i(\sigma))$. It is easy to verify that, for any L, if $\mathrm{Cyl}(L) \in \mathbf{TxtFex}^*(\mathbf{M}_i)$, then $\mathrm{Cyl}(L) \in \mathbf{TxtBc}(\mathbf{M}_{f(i)})$. Part (d) follows.

(e) Let F be a recursive function from SEG to SEG such that, (i) for all $\sigma \subseteq \tau$, $F(\sigma) \subseteq F(\tau)$, and (ii) for all σ, $\mathrm{content}(F(\sigma)) = \{x \mid \langle x, 0\rangle \in \mathrm{content}(\sigma)\}$. Let f be a recursive function such that, for all i and σ, $\mathbf{M}_{f(i)}(\sigma) = g(\mathbf{M}_i(F(\sigma)))$. It is easy to verify that, if $L \in \mathbf{TxtBc}(\mathbf{M}_i)$, then $\mathrm{Cyl}(L) \in \mathbf{TxtBc}(\mathbf{M}_{f(i)})$. Part (e) follows.

(f) Let F be a recursive function from SEG to SEG such that, (i) for all $\sigma \subseteq \tau$, $F(\sigma) \subseteq F(\tau)$, and (ii) for all σ, $\mathrm{content}(F(\sigma)) = \{\langle x, y\rangle \mid x \in \mathrm{content}(\sigma) \wedge y \leq |\sigma|\}$. Let f be a recursive function such that, for all i and σ, $\mathbf{M}_{f(i)}(\sigma) = h_1(\mathbf{M}_i(F(\sigma)))$. It is easy to verify that, if $\mathrm{Cyl}(L) \in \mathbf{TxtBc}(\mathbf{M}_i)$, then $L \in \mathbf{TxtBc}(\mathbf{M}_{f(i)})$. Part (f) follows. ∎

Theorem 8 *NOT* $(\exists f \in \mathcal{R})(\forall x \mid \mathcal{C}_x \in \mathbf{NoisyTxtEx} \cap \mathbf{NoisyInfEx})[\mathcal{C}_x \subseteq \mathbf{TxtBc}(\mathbf{M}_{f(x)})]$.

PROOF. Fix f. By the operator recursion theorem [Cas74, Cas94], there exists a 1-1 increasing recursive function p such that the languages $W_{p(i)}$, $i \geq 0$, are defined as follows. Our construction will ensure that $\mathcal{C}_{p(0)} \in \mathbf{NoisyTxtEx} \cap \mathbf{NoisyInfEx}$. Also, it will be the case that $\mathcal{C}_{p(0)} \not\subseteq \mathbf{TxtBc}(\mathbf{M}_{f(x)})$.

We will use a staging construction to define $W_{p(\cdot)}$. We will start the construction at stage 2 for ease of notation. $W_{p(1)}$ will be a subset of **ODD**, and a member of $\mathcal{C}_{p(0)}$. The construction will use a set O. Informally, O denotes the set of odd numbers we have decided to keep out of $W_{p(1)}$. Let O_s denote O as at the beginning of stage s. Initially, let $O_2 = \emptyset$ (we start at stage 2). Let σ_2 be the empty sequence. Let r_s denote the least odd number not in $O_s \cup (W_{p(1)}$ enumerated before stage $s)$. Enumerate $p(1)$ in $W_{p(0)}$. Go to stage 2.

Stage s

1. Enumerate $p(s)$ into $W_{p(0)}$.

   Enumerate r_s into $W_{p(1)}$.

   Dovetail the execution of steps 2 and 3. If and when step 3 succeeds, go to step 4.
2. Enumerate one-by-one, the elements of $\mathbf{ODD} - O_s - \{r_s\}$ into $W_{p(s)}$.
3. Search for $\tau_s \supset \sigma_s$ and an odd number q_s such that $\mathrm{content}(\tau_s) \subseteq \mathbf{ODD} - (O_s \cup \{r_s\})$ and $q_s \in W_{\mathbf{M}_{f(p(0))}(\tau_s)} - (\{r_s\} \cup O_s \cup \mathrm{content}(\tau_s))$.
4. Enumerate $\mathrm{content}(\tau_s)$ into $W_{p(1)}$.

   Let σ_{s+1} be an extension of τ_s such that $\mathrm{content}(\sigma_{s+1}) = W_{p(1)}$ enumerated until now.

   Enumerate 2 into $W_{p(s)}$.

   Enumerate the (even) number $2s$ into $W_{p(s)}$.

   Let $O_{s+1} = O_s \cup \{q_s\}$, where q_s is as found above in step 3.

   Go to stage $s+1$.

End stage s.

We consider two cases.

Case 1: Stage s starts but does not terminate.

In this case $W_{p(0)} = \{p(i) \mid 1 \leq i \leq s\}$. Note that:

(i) $W_{p(1)}$ is a finite subset of **ODD** containing r_i for $2 \leq i \leq s$.

(ii) for all i, $2 \leq i \leq s-1$, $W_{p(i)}$ is a finite set containing 2 and $2i$ as its only even members.

(iii) for all i, $2 \leq i \leq s-1$, $W_{p(i)}$ does not contain r_i.

(iv) $W_{p(s)}$ is an infinite subset of **ODD** which does not contain r_s.

Thus, $\mathcal{C}_{p(0)}$ is finite, and for each $L, L' \in \mathcal{C}_{p(0)}$, $L \subseteq L'$ implies $L = L'$. Thus, $\mathcal{C}_{p(0)} \in \mathbf{NoisyTxtEx} \cap \mathbf{NoisyInfEx}$ (by Proposition 2).

Let $T \supset \sigma_s$ be a text for $W_{p(s)}$. Now, $(\forall \tau \mid \sigma_s \subset \tau \subset T)[W_{\mathbf{M}_{f(p(0))}(\tau)} \cap \mathbf{ODD}$ is finite] (otherwise step 3 would have succeeded in stage s). Thus, $\mathbf{M}_{f(p(0))}$ does not **TxtBc**-identify $W_{p(s)}$.

Case 2: All stages terminate.

In this case, clearly, for all $i > 1$, $W_{p(i)}$ is finite and contains exactly two even numbers, 2 and $2i$. Also, $W_{p(1)}$ is infinite and contains only odd numbers. The following **M** **NoisyTxtEx**-identifies $\mathcal{C}_{p(0)}$.

$\mathbf{M}(T[n])$

Let $e = \text{card}(\{m < n \mid T(m) \text{ is even}\})$.

If $\text{card}(\text{content}(T[n])) > e$, then output $p(1)$.

Otherwise output $p(j)$ such that $j > 1$ and $\text{card}(\{m < n \mid T(m) = 2j\})$ is maximized.

End

It is easy to verify that $\mathbf{M}$ above **NoisyTxtEx**-identifies $\mathcal{C}_i$. The following $\mathbf{M}$ **NoisyInfEx**-identifies $\mathcal{C}_{p(0)}$.

$\mathbf{M}(I[n])$

If $2 \notin \text{PosInfo}(I[n])$, then output $p(1)$.

Otherwise output $p(j)$ such that $j > 1$ and $j = \min(\{j' \mid j' > 1 \ \wedge \ 2j' \in \text{PosInfo}(I[n])\})$.

End

It is easy to verify that $\mathbf{M}$ above **NoisyInfEx**-identifies $\mathcal{C}_i$.

We now show that $W_{p(1)}$ not in $\mathbf{TxtBc}(\mathbf{M}_{f(p(0))})$.

Let $T = \bigcup_{s \geq 2} \sigma_s$. Clearly, T is a text with content exactly $W_{p(1)}$. Consider any stage $s \geq 2$. It is clear by steps 3 and 4 that, for all s, there exists a τ_s, $\sigma_s \subseteq \tau_s \subseteq \sigma_{s+1}$, such that $q_s \in W_{\mathbf{M}_{f(p(0))}(\tau_s)} - W_{p(1)}$. Thus, $\mathbf{M}_{f(p(0))}$ does not **TxtBc**-identify $W_{p(1)}$.

It follows from the above cases that $\mathcal{C}_{p(0)} \in \mathbf{NoisyTxtEx} \cap \mathbf{NoisyInfEx}$, but $\mathcal{C}_{p(0)} \not\subseteq \mathbf{TxtBc}(\mathbf{M}_{f(p(0))})$. ∎

As a corollary to Theorem 7 we have the following result which implies the impossibility of effective synthesis from enumeration procedures for the noisy inference criteria noted at the beginning of this section.

Corollary 1 *Suppose $n \in N$ and $a \in N \cup \{*\}$. Suppose $\mathbf{J} \in \{$ $\mathbf{NoisyTxtBc}^n$, $\mathbf{NoisyTxtEx}^a$, $\mathbf{NoisyTxtFex}^a$, $\mathbf{NoisyInfBc}^n$, $\mathbf{NoisyInfEx}^a$ $\mathbf{NoisyInfFex}^a$ $\}$. Then,*

NOT $(\exists f \in \mathcal{R})(\forall x \mid \mathcal{C}_x \in \mathbf{J})[\mathcal{C}_x \subseteq \mathbf{J}(\mathbf{M}_{f(x)})]$.

3.2 When effective synthesis is possible

In this section we show that effective synthesis is possible for $\mathbf{NoisyTxtBc}^*$ and $\mathbf{NoisyInfBc}^*$ criteria.

The next theorem allows us to show as a corollary that synthesis of learning machines, from enumeration procedures for r.e. classes of languages, is possible in the case of $\mathbf{NoisyTxtBc}^*$-identification criteria. We also obtain a characterization of $\mathbf{NoisyTxtBc}^*$-identification for r.e. classes in the process (Corollary 3).

Theorem 9 *There exists a recursive function f such that following is satisfied. Suppose $(\forall L, L' \in \mathcal{C}_i \mid L \subseteq L')[L =^* L']$. Then $\mathcal{C}_i \subseteq \mathbf{NoisyTxtBc}^*(\mathbf{M}_{f(i)})$.*

PROOF. $\mathbf{M}_{f(i)}$ is defined as follows. $\mathbf{M}_{f(i)}(T[n]) = \text{Proc}(T[n])$, where $W_{\text{Proc}(T[n])}$ is defined as follows.

$W_{\mathrm{Proc}(T[n])}$

Go to stage 0.

Stage s

Let $w_s = \langle k_s, m_s \rangle$ be the least number such that $k_s \in W_{i,s}$ and $(\forall r \mid m_s \le r < n)[T(r) \in W_{k_s,s}]$.

Enumerate $W_{k_s,s}$ in $W_{\mathrm{Proc}(T[n])}$.

End Stage s

End

Now suppose T is a noisy text for $L \in \mathcal{C}_i$. Let $w = \langle k, m \rangle$ be the least number such that $k \in W_i$ and $(\forall r \ge m)[T(r) \in W_k]$. Note that there exists such a $w = \langle k, m \rangle$. Also for such a $w = \langle k, m \rangle$, $L \subseteq W_k$ (since T is a noisy text for L). Let n_0 be such that for all $w' = \langle k', m' \rangle$, where $w' < w$ and $k' \in W_i$, there exists an r such that $m' \le r < n_0$ and $T(r) \notin W_{k'}$. It follows that for all $n \ge n_0$, for all but finitely many s, the w_s as computed in stage s of $W_{\mathrm{Proc}(T[n])}$ is w. Thus, $W_{\mathrm{Proc}(T[n])} =^* W_k$. Since $L \subseteq W_k$, it follows from the hypothesis of the theorem that $L =^* W_k$. Thus, $W_{\mathrm{Proc}(T[n])} =^* L$. Hence, $\mathbf{M}_{f(i)}$ $\mathbf{TxtBc}^*$-identifies $\mathcal{C}_i$. ∎

Theorems 9 and 1 imply the following corollaries. The first provides a positive synthesis result, the second a corresponding characterization which is *a* strict subset principle.

Corollary 2 $(\exists f \in \mathcal{R})[\mathcal{C}_i \in \mathbf{NoisyTxtBc}^* \Rightarrow \mathcal{C}_i \subseteq \mathbf{NoisyTxtBc}^*(\mathbf{M}_{f(i)})]$.

Corollary 3 $\mathcal{C}_i \in \mathbf{NoisyTxtBc}^* \Leftrightarrow (\forall L, L' \in \mathcal{C}_i \mid L \subseteq L')[L =^* L']$.

The next theorem allows us to show as a corollary that synthesis of learning machines, from enumeration procedures for r.e. classes of languages, is possible in the case of $\mathbf{NoisyInfBc}^*$-identification. We also obtain a characterization of $\mathbf{NoisyInfBc}^*$-identification for r.e. classes in the process (Corollary 5).

Theorem 10 *There exists a recursive function f such that the following is satisfied. Suppose for all $L \in \mathcal{C}_i$, there exists an $n \in N$ such that $(\forall L' \in \mathcal{C}_i \mid \{x \in L \mid x \le n\} = \{x \in L' \mid x \le n\})[L =^* L']$. Then $\mathcal{C}_i \in \mathbf{NoisyInfBc}^*(\mathbf{M}_{f(i)})$.*

PROOF. $\mathbf{M}_{f(i)}$ is defined as follows. $\mathbf{M}_{f(i)}(I[m]) = \mathrm{Proc}(I[m])$, where $\mathrm{Proc}(I[m])$ is defined as follows.

$W_{\mathrm{Proc}(I[m])}$

Let $Pos = \mathrm{PosInfo}(I[m])$.

Let $Neg = \mathrm{NegInfo}(I[m])$.

Go to stage 0.

Begin stage s

For each $j \in W_{i,m}$, let $\mathrm{match}(j, s) = \min(W_{j,s} \boldsymbol{\Delta} Pos)$.

Let j^s denote $j \in W_{i,m}$ which maximizes $\mathrm{match}(j, s)$.

Enumerate $W_{j^s,s}$.

End stage s

End

Now suppose I is a noisy informant for W_j, where $j \in W_i$. Let n be such that $(\forall L' \in \mathcal{C}_i \mid \{x \in L \mid x \leq n\} = \{x \in L' \mid x \leq n\})[L =^* L']$. Let m_0 be so large that, for all $m \geq m_0$, $I(m) \notin \{(x, 1-\chi_L(x)) \mid x \leq n\}$ and $\mathrm{PosInfo}(I[m_0]) = \{x \in W_j \mid x \leq n\}$. Moreover, assume that $j \in W_{i,m_0}$.

Now consider the computation of $\mathrm{Proc}(I[m])$ for any $m \geq m_0$. Since $\mathrm{match}(j,s) \geq n$, for large enough s, $\lim_{s\to\infty} j^s$ converges to a j' such that $\{x \in W_{j'} \mid x \leq n\} = \{x \in W_j \mid x \leq n\}$. It thus follows from the hypothesis that $W_j =^* W_{j'} =^* W_{\mathrm{Proc}(T[m])}$. Thus, $\mathbf{M}_{f(i)}$ $\mathbf{NoisyInfBc}^*$-identifies W_j. Since j was an arbitrary member of W_i, we have that $\mathbf{M}_{f(i)}$ $\mathbf{NoisyInfBc}^*$-identifies $\mathcal{C}_i$. ∎

As corollaries to Theorem 2 and Theorem 10 we have the following two corollaries. The first provides a positive synthesis result, and the second a corresponding characterization which is a kind of informant analog of a subset principle.

Corollary 4 $(\exists f \in \mathcal{R})[\mathcal{C}_i \in \mathbf{NoisyInfBc}^* \Rightarrow \mathcal{C}_i \subseteq \mathbf{NoisyInfBc}^*(\mathbf{M}_{f(i)})]$.

Corollary 5 $\mathcal{C}_i \in \mathbf{NoisyInfBc}^*$ *iff* $(\forall L \in \mathcal{C}_i)(\exists n \in N)(\forall L' \in \mathcal{C}_i \mid \{x \in L \mid x \leq n\} = \{x \in L' \mid x \leq n\})[L =^* L']$.

In (1) from Section 1, the finite sets S are called *tell-tales* [Ang80]. Essentially the n in Corollary 5 just above defines a finite initial segment of an informant which is the informant-analog of a tell-tale.

4 Identification from uniform decision procedures

In this section we show that effective synthesis of learning machines, from decision procedures for indexed families of recursive languages, is possible for the following noisy inference criteria.

- $\mathbf{NoisyTxtBc}^a$, for $a \in N \cup \{*\}$;
- $\mathbf{NoisyInfBc}^a$, for $a \in N \cup \{*\}$;
- **NoisyTxtEx**;
- $\mathbf{NoisyTxtEx}^*$;
- **NoisyTxtFex**;
- $\mathbf{NoisyTxtFex}^*$;
- $\mathbf{NoisyInfEx}^n$, for $n \in N$; and
- $\mathbf{NoisyInfFex}^n$, for $n \in N$.

In the process we give a characterization of the above criteria for indexed families of recursive languages in terms of variants of the subset principle. We are also able to show that effective synthesis from decision procedures for indexed families *is* possible for $\mathbf{NoisyTxtEx}^n$ (for $n \in N$) and $\mathbf{NoisyTxtFex}^n$ (for $n \in N$) *if we allow doubling of errors* (see Corollary 7 below). However, for $\mathbf{NoisyInfEx}^*$ and $\mathbf{NoisyInfFex}^*$, effective synthesis is not possible.

We first consider effective synthesis from decision procedures for indexed families in the context of inference criteria involving noisy texts. This is followed by similar treatment of inference criteria involving noisy informants.

4.1 Effective synthesis for noisy text inference criteria

As noted above effective synthesis from decision procedures for indexed families is possible for all noisy text inference criteria except $\mathbf{NoisyTxtEx}^n$ (for $n \in N$) and $\mathbf{NoisyTxtFex}^n$ (for $n \in N$). In Section 4.1.1 we first establish the positive results where effective synthesis is possible. These results will also show that in the cases in which effective synthesis is not possible, we can get a weaker form of effective synthesis if we are willing to tolerate up to twice the number of errors in the final grammar. In Section 4.1.2, we establish that this is the best we can do.

4.1.1 When effective synthesis is possible

Theorem 11 *There exists a recursive function f such that following is satisfied.*
$(\forall i \mid \mathcal{U}_i \neq \emptyset)(\forall j \in W_i)(\forall \text{ noisy texts } T \text{ for } U_j)[\mathbf{M}_{f(i)}(T)\downarrow = k \in W_i \text{ such that } U_k \supseteq U_j]$.

PROOF. The idea of the proof of this theorem is similar to that of Theorem 9. Define $\mathbf{M}_{f(i)}$ as follows. $\mathbf{M}_{f(i)}$ on a text T converges to a $k \in W_i$, if any, such that there exists an m, $(\forall n \geq m)[\varphi_k(T(n))\uparrow \vee \varphi_k(T(n)) = 1]$. Note that $\mathbf{M}_{f(i)}$ behaving as above can be constructed effectively from i. It is easy to verify that, if

(1) $\mathcal{U}_i$ is not empty (thus, in particular, for all $j \in W_i$, j is a decision procedure), and

(2) T is a noisy text for U_j, $j \in W_i$,

then $\mathbf{M}_{f(i)}$ on T converges to a $k \in W_i$ such that $U_j \subseteq U_k$. ∎

The above theorem implies the following corollary.

Corollary 6 *There exists a recursive function f such that following is satisfied. Suppose $(\forall L, L' \in \mathcal{U}_i \mid L \subseteq L')[L =^a L']$. Then, $\mathcal{U}_i \subseteq \mathbf{NoisyTxtEx}^a(\mathbf{M}_{f(i)})$.*

As a corollary, using Theorem 1 we have the following.

Corollary 7 $(\exists f \in \mathcal{R})(\forall a \in N \cup \{*\})(\forall i)[\mathcal{U}_i \in \mathbf{NoisyTxtBc}^a \Rightarrow \mathcal{U}_i \subseteq \mathbf{NoisyTxtEx}^{2a}(\mathbf{M}_{f(i)})]$.

As a corollary, using Theorem 1 we have that effective synthesis, from decision procedures for indexed families, is possible for **NoisyTxtEx**, **NoisyTxtBc**, $\mathbf{NoisyTxtEx}^*$ and $\mathbf{NoisyTxtBc}^*$ criteria. Further, we get a characterization of the above criteria as shown in the following two corollaries. The first shows that, *for indexed families*, **NoisyTxtBc** collapses to **NoisyTxtEx** and they are characterized by a strict subset principle. The second is similar, but for $\mathbf{NoisyTxtBc}^*$ and $\mathbf{NoisyTxtEx}^*$.

Corollary 8 $(\forall i)[\mathcal{U}_i \in \mathbf{NoisyTxtBc} \Leftrightarrow \mathcal{U}_i \in \mathbf{NoisyTxtEx} \Leftrightarrow (\forall L, L' \in \mathcal{U}_i)[L \subseteq L' \Rightarrow L = L']]$.

Corollary 9 $(\forall i)[\mathcal{U}_i \in \mathbf{NoisyTxtBc}^* \Leftrightarrow \mathcal{U}_i \in \mathbf{NoisyTxtEx}^* \Leftrightarrow (\forall L, L' \in \mathcal{U}_i)[L \subseteq L' \Rightarrow L =^* L']]$.

The following theorem is used to show that effective synthesis from decision procedures is possible for $\mathbf{NoisyTxtBc}^a$-identification, for $a \in N \cup \{*\}$. We also get a characterization of $\mathbf{NoisyTxtBc}^a$ in the process.

Theorem 12 *Suppose $a \in N$. There exists a recursive function g such that following is satisfied. Suppose $[(\forall L \in \mathcal{U}_i)(\forall L' \in \mathcal{U}_i \mid L \subseteq L')[L =^{2a} L']]$. Then, $\mathcal{U}_i \subseteq$ $\mathbf{NoisyTxtBc}^a(\mathbf{M}_{g(i)})$.*

Proof. The idea of the proof is to use Theorem 11 along with a modification of the trick used by Case and Lynes [CL82] to show that $\mathbf{TxtEx}^{2a} \subseteq \mathbf{TxtBc}^a$. Let f be as given by Theorem 11. Suppose i is as given in the hypothesis. Note that for any noisy text T for U_j, $j \in W_i$, $\mathbf{M}_{f(i)}(T)$ converges to $k \in W_i$ such that $U_k \supseteq U_j$. Let g be a recursive function such that $\mathbf{M}_{g(i)}(T[n]) = \mathrm{Proc}(T[n])$, where $\mathrm{Proc}(T[n])$ is defined as follows. For ease of notation, in the following we assume that, for all $j \in W_i$, j is a decision procedure. This is fine, since if some $j \in W_i$ is not a decision procedure, then it doesn't matter what $\mathrm{Proc}(T[n])$ does.

$W_{\mathrm{Proc}(T[n])}$

Let $k = \mathbf{M}_{f(i)}(T[n])$.

Let $X = U_k[n]$.

For each $x \in X$, let $o^n(x) = x + \mathrm{occur}(T[n], x)$.

Let $x_0^n, x_1^n, \ldots, x_{\mathrm{card}(X)-1}^n$ be the sorting of elements of X based on non-decreasing order of $o^n(x)$, where ties are broken based on values of x_i^n (i.e., for $i < \mathrm{card}(X) - 1$, $o^n(x_i^n) \leq o^n(x_{i+1}^n)$, and if $o^n(x_i^n) = o^n(x_{i+1}^n)$, then $x_i^n < x_{i+1}^n$).

Let $S_n = \{x_i^n \mid i < a\}$.

Let $W_{\mathrm{Proc}(T[n])} = U_k - S_n$.

End

Now suppose every member of W_i is a decision procedure, let $j \in W_i$, and let T be a noisy text for U_j. Suppose $k = \mathbf{M}_{f(i)}(T)$. Note that $U_j \subseteq U_k$. Moreover, $U_j =^{2a} U_k$. Let $Y = U_k - U_j$. Note that for all $x \in U_j$, $\lim_{s\to\infty} o^s(x) = \infty$; however, for $x \notin U_j$, $\lim_{s\to\infty} o^s(x)\downarrow < \infty$. Thus, $x_0^n, x_1^n, \ldots, x_{\mathrm{card}(Y)-1}^n$ converge (as $n \to \infty$) to the different elements of Y. (Note that we needed to add x in the definition of $o^n(x)$ to ensure that the non-occurrence of large numbers in initial segments of T does not spoil this property).

We consider two cases,

Case 1: $\mathrm{card}(Y) \leq a$.

In this case, for all but finitely many n, $Y \subseteq S_n$. Thus, for large enough n, $W_{\mathrm{Proc}(T[n])} \subseteq U_j$, and $\mathrm{card}(U_j - W_{\mathrm{Proc}(T[n])}) = a - \mathrm{card}(Y)$. It follows that $\mathbf{M}$ $\mathbf{NoisyTxtBc}^a$-identifies U_j.

Case 2: $\mathrm{card}(Y) > a$.

Note that $W_{\mathrm{Proc}(T[n])} = U_k - \{x_0^n, x_1^n, \ldots, x_{a-1}^n\}$. Also, $\mathrm{card}(Y) > a$, $x_0^n, x_1^n, \ldots, x_{a-1}^n$ converge (as $n \to \infty$) to a different elements of Y. Since $U_j = U_k - Y$ and $\mathrm{card}(Y) \leq 2a$, it follows that for large enough n, $W_{\mathrm{Proc}(T[n])} =^a U_j$. Thus, $\mathbf{M}$ $\mathbf{NoisyTxtBc}^a$-identifies U_j. ∎

As a corollary to Theorem 12, using Theorem 1 and Corollary 7, we have

Corollary 10 *Suppose $a \in N \cup \{*\}$. $(\exists g \in \mathcal{R})[\mathcal{U}_i \in \mathbf{NoisyTxtBc}^a \Rightarrow \mathcal{U}_i \subseteq \mathbf{NoisyTxtBc}^a(\mathbf{M}_{g(i)})]$.*

Hence, effective synthesis from decision procedures is possible for $\mathbf{NoisyTxtBc}^a$-identification.

As another corollary to Theorem 12, using Theorem 1 and Corollary 9, we have the following strict subset principle characterization of $\mathbf{NoisyTxtBc}^a$.

Corollary 11 *Suppose $a \in N \cup \{*\}$. $\mathcal{U}_i \in \mathbf{NoisyTxtBc}^a \Leftrightarrow [(\forall L \in \mathcal{U}_i)(\forall L' \in \mathcal{U}_i \mid L' \subseteq L)[L' =^{2a} L]]$.*

4.1.2 When effective synthesis is not possible

We have already seen that effective synthesis is possible for $\mathbf{NoisyTxtEx}^n$ ($n \in N$) and for $\mathbf{NoisyTxtFex}^n$ ($n \in N$) if we are willing to tolerate up to $2n$ number of errors in the final grammar(s). We next show that this is the best possible effective synthesis result in these two cases.

Theorem 13 *Suppose $n \in N$, and $n > 0$. NOT $(\exists f \in \mathcal{R})(\forall i \mid \mathcal{U}_i \in \mathbf{NoisyTxtEx}^n)[\mathcal{U}_i \subseteq \mathbf{TxtEx}^{2n-1}(\mathbf{M}_{f(i)})]$.*

PROOF. Fix $n \in N$ and $f \in \mathcal{R}$. Then by the operator recursion theorem, there exists a recursive, one-to-one, increasing function p such that $\varphi_{p(i)}$ may be defined as follows. Intuitively, $W_{p(0)}$ will enumerate a subset of $\{p(1), p(2), \ldots\}$. It will be the case that $W_{p(0)}$ is non-empty, and for all $p(j) \in W_{p(0)}$, $p(j)$ is a decision procedure. Let $\varphi_{p(1)}$ be a characteristic function for **ODD**. Let σ_2 be an empty sequence (we start from stage 2 for ease of notation). Let $x_2 = 0$. Intuitively x_s bounds the even numbers used in the diagonalization in stages numbered $< s$. Enumerate $p(1)$ in $W_{p(0)}$. Go to stage 2.

Stage s

1. Search for an extension τ of σ_s and a set $S \subseteq \mathbf{ODD}$ of cardinality $2n$ such that content$(\tau) \subseteq$ **ODD** and $W_{\mathbf{M}_{f(p(0))}(\sigma_s)} - \text{content}(\tau) \supseteq S$.
2. Let τ, S be as found in step 1. Dovetail steps 3 and 4, until step 3 succeeds. If and when step 3 succeeds, go to step 5.
3. Search for an extension τ' of τ such that content$(\tau') \subseteq$ **ODD** and $\mathbf{M}_{f(p(0))}(\tau) \neq \mathbf{M}_{f(p(0))}(\tau')$.
4. Enumerate $p(s)$ in $W_{p(0)}$.

   For $x = 0$ to ∞ do

   If x is even or $x \in S$, then let $\varphi_{p(s)}(x) = 0$;

   Else let $\varphi_{p(s)}(x) = 1$.

   EndFor
5. Let x be the least number such that $\varphi_{p(s)}(x)$ has not been defined until now. Let y be the least even number $> \max(\{x, x_s\})$.

   Let $\varphi_{p(s)}(y) = 1$.

   For all $z \geq x$ such that $z \neq y$, let $\varphi_{p(s)}(z) = 0$.

   Let $x_{s+1} = y$.

   Let σ_{s+1} be an extension of τ' such that content$(\sigma_{s+1}) \supseteq \{2x + 1 \mid x \leq s\}$.

   Go to stage $s + 1$.

End stage s

It is easy to verify that for all $p(j) \in W_{p(0)}$, $p(j)$ is a decision procedure. We now consider two cases.

Case 1: All stages terminate.

In this case, $W_{p(0)} = \{p(j) \mid j \geq 1\}$. Note that (a) $U_{p(1)} =$ **ODD**, (b) for each $s > 1$, $U_{p(s)}$ is finite and contains exactly one even number x_{s+1}, and (c) x_s's are pairwise distinct. It follows that $\mathcal{U}_{p(0)} \in \mathbf{NoisyTxtEx}$. However, $T = \bigcup_{s \in N} \sigma_s$ is a text for **ODD** on which $\mathbf{M}_{f(p(0))}$ makes infinitely many mind changes.

Case 2: Stage s starts but does not terminate.

Case 2.1: In stage s, step 1 does not succeed.

In this case $W_{p(0)} = \{p(j) \mid 1 \leq j < s\}$. Note that (a) $U_{p(1)} = \mathbf{ODD}$, (b) for $1 < j < s$, $U_{p(j)}$ is finite and contains exactly one even number x_{j+1}, and (c) x_j's are pairwise distinct. It follows that $\mathcal{U}_{p(0)} \in \mathbf{NoisyTxtEx}$. However, for all τ such that $\sigma_s \subseteq \tau$ and content$(\tau) \subseteq \mathbf{ODD}$, $W_{\mathbf{M}_{f(p(0))}(\tau)}$ contains at most finitely many odd numbers. Thus, $\mathbf{M}_{f(p(0))}$ does not $\mathbf{TxtEx}^{2n-1}$-identify $\mathcal{U}_{p(0)}$.

Case 2.2: In stage s, step 1 succeeds, but step 3 does not succeed.

In this case $W_{p(0)} = \{p(j) \mid 1 \leq j \leq s\}$. Note that (a) $U_{p(1)} = \mathbf{ODD}$, (b) for $1 < j < s$, $U_{p(j)}$ is finite and contains exactly one even number x_{j+1}, (c) x_j are pairwise distinct, and (d) $U_{p(s)}$ is $\mathbf{ODD}-S$, where S is as in step 2 of stage s and card$(S) = 2n$. It follows that $\mathcal{U}_{p(0)} \in \mathbf{NoisyTxtEx}^n$. Let τ be as in step 2 of stage s. Now, content$(\tau) \subseteq U_{p(s)}$, $S \subseteq W_{M_{f(p(0))}(\tau)}$, and for all $\tau' \supseteq \tau$ such that content$(\tau') \subseteq \mathbf{ODD}$, $\mathbf{M}_{f(p(0))}(\tau) = \mathbf{M}_{f(p(0))}(\tau')$. Thus, $\mathbf{M}_{f(p(0))}$ does not $\mathbf{TxtEx}^{2n-1}$-identify $W_{p(s)} \in \mathcal{U}_{p(0)}$.

From the above cases we have that $\mathbf{M}_{f(p(0))}$ does not $\mathbf{TxtEx}^{2n-1}$-identify $\mathcal{U}_{p(0)} \in \mathbf{NoisyTxtEx}^n$. This proves the theorem. ∎

The above proof can be generalized to show the following result.

Theorem 14 *Suppose $n \in N$, and $n > 0$. NOT* $(\exists f \in \mathcal{R})(\forall i \mid \mathcal{U}_i \in \mathbf{NoisyTxtEx}^n)[\mathcal{U}_i \subseteq \mathbf{TxtFex}^{2n-1}(\mathbf{M}_{f(i)})]$.

Proof. Fix $n \in N$ and $f \in \mathcal{R}$. Then by the operator recursion theorem, there exists a recursive 1–1 increasing function p such that $\varphi_{p(i)}$ may be defined as follows. Intuitively, $W_{p(0)}$ will enumerate a subset of $\{p(1), p(2), \ldots\}$. It will be the case that $W_{p(0)}$ is non-empty, and for all $p(j) \in W_{p(0)}$, $p(j)$ is a decision procedure. Let $\varphi_{p(1)}$ be a characteristic function for $\mathbf{ODD}$. Enumerate $p(1)$ in $W_{p(0)}$. Let σ_0 be such that content$(\sigma_0) = \{1\}$. Let $l_0 = 0$. We will always have content$(\sigma_s) = \{2i+1 \mid i \leq l_s\}$. Let curx $= 0$. Intuitively curx bounds the even numbers used earlier in the diagonalization. Let curprog $= 1$. Intuitively, curprog denotes the maximum i such that $p(i)$ has been used in diagonalization. Go to stage 0.

Stage s

1. Let $P = \text{ProgSet}(\mathbf{M}_{f(p(0))}, \sigma_s)$.
   Let $X = \{2i+1 \mid l_s < i \leq l_s + 4n * (\text{card}(P)+1)\}$.
   Go to substage 0.
2. Substage s'
   3. Let $Q = \{q \in P \mid \text{card}(X - W_{q,s'}) \leq 4n\}$.
      Let Y be a subset of X, of cardinality $2n$, such that, for all $q \in Q$, $W_{q,s'} \supseteq Y$.
      Let curprog = curprog + 1.
      Enumerate $p(\text{curprog})$ in $W_{p(0)}$.
      Dovetail steps 4, 5 and 6 until step 4 or 5 succeeds. If step 4 succeeds (before step 5 succeeds, if ever), then go to step 8. If step 5 succeeds (before step 4 succeeds, if ever), then go to step 7.
   4. Search for an extension σ' of σ_s such that content$(\sigma') \subseteq \mathbf{ODD}$ and $\text{ProgSet}(\mathbf{M}_{f(p(0))}, \sigma') \neq \text{ProgSet}(\mathbf{M}_{f(p(0))}, \sigma_s)$.
   5. Search for a $q \in P - Q$ such that card$(X - W_q) \leq 4n$.

6. For $x = 0$ to ∞

   If x is even or $x \in Y$, then let $\varphi_{p(\mathrm{curprog})}(x) = 0$;

   Otherwise let $\varphi_{p(\mathrm{curprog})}(x) = 1$.

   Endfor

7. Let x be the least number such that $\varphi_{p(\mathrm{curprog})}(x)$ has not been defined until now. Let y be the least even number $> \max(\{x, \mathrm{curx}\})$.

   Let $\varphi_{p(\mathrm{curprog})}(y) = 1$.

   For all $z \geq x$ such that $z \neq y$, let $\varphi_{p(\mathrm{curprog})}(z) = 0$.

   Let $\mathrm{curx} = y$.

   Go to substage $s' + 1$.

End substage s'

8. Let x be the least number such that $\varphi_{p(\mathrm{curprog})}(x)$ has not been defined until now. Let y be the least even number $> \max(\{x, \mathrm{curx}\})$.

   Let $\varphi_{p(\mathrm{curprog})}(y) = 1$.

   For all $z \geq x$ such that $z \neq y$, let $\varphi_{p(\mathrm{curprog})}(z) = 0$.

   Let $\mathrm{curx} = y$.

   Let $l_{s+1} = 2 + (\max(\mathrm{content}(\sigma')) - 1)/2$.

   Let σ_{s+1} be an extension of σ' such that $\mathrm{content}(\sigma_{s+1}) = \{2x + 1 \mid x \leq l_{s+1}\}$.

   Go to stage $s + 1$.

End stage s

It is easy to verify that for all $p(j) \in W_{p(0)}$, $p(j)$ is a decision procedure. We now consider the following cases.

Case 1: All stages terminate.

In this case, $W_{p(0)} = \{p(j) \mid j \geq 1\}$. Note that (a) $U_{p(1)} = \mathbf{ODD}$, (b) for each $i > 1$, $U_{p(i)}$ is finite and contains exactly one even number, and (c) the even number in $U_{p(i)}$, $i > 1$, are pairwise different. It follows that $\mathcal{U}_{p(0)} \in \mathbf{NoisyTxtEx}$. However, $T = \bigcup_{s \in N} \sigma_s$ is a text for $\mathbf{ODD} \in \mathcal{U}_{p(0)}$, but $\mathrm{ProgSet}(\mathbf{M}_{f(p(0))}, T)$ is infinite.

Case 2: Stage s starts but does not terminate.

First note that there cannot be infinitely many substages in stage s (since Q takes a limiting value). Let substage s' be last substage which is executed. Let curprog, P, Q, Y below denote the values of these variables at the end of step 3 of substage s' in stage s.

Note that in this case $W_{p(0)} = \{p(j) \mid 1 \leq j \leq \mathrm{curprog}\}$. Also, (a) $U_{p(1)} = \mathbf{ODD}$, (b) for $1 < j < \mathrm{curprog}$, $U_{p(j)}$ is finite and contains exactly one even number, (c) the even number in $U_{p(j)}$, $1 < j < \mathrm{curprog}$ are pairwise distinct, and (d) $U_{p(\mathrm{curprog})}$ is $\mathbf{ODD} - Y$, where Y is as defined in substage s' of stage s. Note that $\mathrm{card}(Y) = 2n$. It follows that $\mathcal{U}_{p(0)} \in \mathbf{NoisyTxtEx}^n$.

Also, due to non success of steps 4, 5 in substage s' of stage s, it follows that for all $\sigma \supseteq \sigma_s$ such that $\mathrm{content}(\sigma) \subseteq \mathbf{ODD}$, (a) $\mathrm{ProgSet}(\mathbf{M}_{f(p(0))}, \sigma) = P$, (b) $\forall q \in Q$, $W_q - W_{p(\mathrm{curprog})} \supseteq Y$, (c) $\forall q \in P - Q$, $\mathrm{card}(\mathbf{ODD} - W_q) \geq 4n$. It follows that $\mathbf{M}_{f(p(0))}$ does not $\mathbf{TxtFex}^{2n-1}$-identify $W_{p(\mathrm{curprog})}$.

From the above cases, it follows that $\mathbf{M}_{f(p(0))}$ does not $\mathbf{TxtFex}^{2n-1}$-identify $\mathcal{U}_{p(0)} \in \mathbf{NoisyTxtFex}^n$. This proves the theorem. ▮

4.2 Effective synthesis for noisy informant inference criteria

We now turn our attention to effective synthesis from uniform decision procedures for indexed families in the context of noisy informant inference criteria. We first consider cases where effective synthesis is possible, followed by those cases where effective synthesis is not possible.

4.2.1 When effective synthesis is possible

We first consider $\mathbf{NoisyInfEx}^a$-identification for $a \in N$.

Theorem 15 *Suppose $a \in N$. There exists a recursive function f such that the following is satisfied. Suppose for all $L \in \mathcal{U}_i$, there exists an n such that $(\forall L' \in \mathcal{U}_i \mid \{x \in L \mid x \leq n\} = \{x \in L' \mid x \leq n\})[L =^a L']$. Then $\mathcal{U}_i \subseteq \mathbf{NoisyInfEx}^a(\mathbf{M}_{f(i)})$*

Proof. Suppose the hypothesis. Suppose I is a noisy informant for $L \in \mathcal{U}_i$. Let Gram(j) denote a grammar, effectively obtained from j, for $\{x \mid \varphi_j(x) = 1\}$. $\mathbf{M}_{f(i)}$ on I searches for $\langle j, n, m\rangle$ such that

(a) $j \in W_i$,

(b) $(\forall j' \in W_i \mid U_j[n] = U_{j'}[n])[U_j =^a U_{j'}]$, and

(c) $(\forall m' \geq m)[I(m) \notin \{(x, 1 - U_j(x)) \mid x < n\}]$.

Note that such a $\langle j, n, m\rangle$, if any, can be found in the limit. $\mathbf{M}_{f(i)}$ then outputs, on input I, Gram(j) in the limit. It is easy to verify using the hypothesis that, for all noisy informants I for $L \in \mathcal{U}_i$, there exists a $\langle j, n, m\rangle$ satisfying (a), (b), and (c) above. Clearly, any $\langle j, n, m\rangle$ satisfying (a), (b), and (c) above also has the property that $W_{\mathrm{Gram}(j)} =^a L$. Thus $\mathbf{M}_{f(i)}$ $\mathbf{NoisyInfEx}^a$-identifies $\mathcal{U}_i$. ∎

As a corollary, using Theorem 4 and Theorem 5, we have the following informant-style tell-tale characterization.

Corollary 12 $(\forall a \in N)(\forall i \mid \mathcal{U}_i \neq \emptyset)[\mathcal{U}_i \in \mathbf{NoisyInfEx}^a \Leftrightarrow \mathcal{U}_i \in \mathbf{NoisyInfFex}^a \Leftrightarrow (\forall L \in \mathcal{U}_i)(\exists n)(\forall L' \in \mathcal{U}_i \mid \{x \in L \mid x \leq n\} = \{x \in L' \mid x \leq n\})[L =^a L']]$.

As corollaries to Theorems 15 and 5 we get the following positive results about effective synthesis for $\mathbf{NoisyInfEx}^a$ and $\mathbf{NoisyInfFex}^a$-identification, for $a \in N$.

Corollary 13 $(\forall a \in N)(\exists f \in \mathcal{R})(\forall i \mid \mathcal{U}_i \in \mathbf{NoisyInfEx}^a)[\mathcal{U}_i \subseteq \mathbf{NoisyInfEx}^a(\mathbf{M}_{f(i)})]$.

Corollary 14 $(\forall a \in N)(\exists f \in \mathcal{R})(\forall i \mid \mathcal{U}_i \in \mathbf{NoisyInfFex}^a)[\mathcal{U}_i \subseteq \mathbf{NoisyInfFex}^a(\mathbf{M}_{f(i)})]$.

As a corollary to Theorems 15 and 2 we have

Corollary 15 *Suppose $a \in N$.* $(\exists f \in \mathcal{R})(\forall i \mid \mathcal{U}_i \in \mathbf{NoisyInfBc}^a)[\mathcal{U}_i \subseteq \mathbf{NoisyInfEx}^{2a}(\mathbf{M}_{f(i)})]$.

Since, from a machine $\mathbf{M}$, one can effectively construct a machine $\mathbf{M}'$ which $\mathbf{NoisyInfBc}^a$-identifies $\mathbf{NoisyInfEx}^{2a}(\mathbf{M})$ (see [CJS96]), we immediately have (using Corollary 4 for the $*$-case) the following result about effective synthesis for $\mathbf{NoisyInfBc}^a$-identification.

Corollary 16 *Suppose $a \in N \cup \{*\}$.* $(\exists f \in \mathcal{R})(\forall i \mid \mathcal{U}_i \in \mathbf{NoisyInfBc}^a)[\mathcal{U}_i \subseteq \mathbf{NoisyInfBc}^a(\mathbf{M}_{f(i)})]$.

The following corollary provides an informant-style tell-tale characterization of $\mathbf{NoisyInfBc}^a$ for indexed families of recursive languages.

Corollary 17 *Suppose* $a \in N \cup \{*\}$. $(\forall i \mid \mathcal{U}_i \neq \emptyset)[\mathcal{U}_i \in \mathbf{NoisyInfBc}^a \Leftrightarrow (\forall L \in \mathcal{U}_i)(\exists n)(\forall L' \in \mathcal{U}_i \mid \{x \in L \mid x \leq n\} = \{x \in L' \mid x \leq n\})[L =^{2a} L']]$.

4.2.2 When effective synthesis is not possible

Since $\mathbf{NoisyInfEx}^* \not\subseteq \mathbf{TxtBc}^n$, we have

Theorem 16 *NOT* $(\exists f \in \mathcal{R})(\exists n \in N)(\forall x \mid \mathcal{U}_x \in \mathbf{NoisyInfEx}^*)[\mathcal{U}_x \subseteq \mathbf{TxtBc}^n(\mathbf{M}_{f(x)})]$.

The following theorem shows that effective synthesis, from decision procedures, cannot be done in the case of $\mathbf{NoisyInfEx}^*$-identification.

Theorem 17 *NOT* $(\exists f \in \mathcal{R})(\forall x \mid \mathcal{U}_x \in \mathbf{NoisyInfEx}^*)[\mathcal{U}_x \subseteq \mathbf{TxtFex}^*(\mathbf{M}_{f(x)})]$.

Proof. Fix f. By the operator recursion theorem, there exists a 1-1 increasing recursive function p such that $W_{p(0)}$, and $\varphi_{p(i)}$, $i \geq 1$, are defined as follows.

For all $x \in N$, $\varphi_{p(1)}(x) = 1$. Enumerate $p(1)$ in $W_{p(0)}$.

We will use a staging construction to define $W_{p(j)}$, for $j > 1$. Let σ_0 be the empty sequence, and $x_0 = 0$ (intuitively, x_s is such that $\text{content}(\sigma_s) = \{x \mid x < x_s\}$). Let $j_0 = 2$. Intuitively, j_s denotes the least j such that $p(j)$ has not been used for diagonalization before stage s. Go to stage 0.

Stage s

1. Let $S = \text{ProgSet}(\mathbf{M}_{f(p(0))}), \sigma_s)$.
   Let $t = \text{card}(S)$.
   For $j_s \leq j \leq j_s + t$, enumerate j in $W_{p(0)}$.
   For $j_s \leq j \leq j_s + t$ and for $x < x_s$, let $\varphi_{p(j)}(x) = 1$.
   Dovetail steps 2 and 3, until step 2 succeeds. If and when step 2 succeeds, go to step 4.
2. Search for an extension τ of σ_s such that $\text{ProgSet}(\mathbf{M}_{f(p(0))}, \tau) \neq \text{ProgSet}(\mathbf{M}_{f(p(0))}, \sigma_s)$.
3. For $x = x_s$ to ∞ do
   Let $k = x \bmod (t + 1)$.
   Let $\varphi_{p(j_s+k)}(x) = 1$.
   For $k' < t + 1$ such that $k' \neq k$, let $\varphi_{p(j_s+k')}(x) = 0$.
   EndFor
4. Let τ be as in step 2. Let $x_{s+1} = 2 + \max(\text{content}(\tau))$. Let σ_{s+1} be an extension of τ such that $\text{content}(\sigma_{s+1}) = \{x \mid x < x_{s+1}\}$.
   For $j_s \leq j \leq j_s + t$ and for y such that $\varphi_{p(j)}(y)$ has not been defined until now, let $\varphi_{p(j)}(y) = 1$.
   (Note that $U_{p(j)}$ is thus a finite variant of N).
   Let $j_{s+1} = j_s + t + 1$.
   Go to stage $s + 1$.

End stage s.

We now consider two cases.

Case 1: All stages terminate.

In this case, clearly, $W_{p(0)} = \{p(j) \mid j \geq 1\}$ and all $U_{p(j)}$, $j \geq 1$, are finite variants of N. Thus, $\mathcal{U}_{p(0)} \in \mathbf{NoisyInfEx}^*$. Let $T = \bigcup_{s \in N} \sigma_s$, a text for $W_{p(1)} = N \in \mathcal{U}_{p(0)}$. Now, $\text{ProgSet}(\mathbf{M}_{f(p(0))}, T)$ is infinite. Thus, $\mathbf{M}_{f(p(0))}$ does not $\mathbf{TxtFex}^*$-identify $\mathcal{U}_{p(0)}$.

Case 2: Stage s starts but does not terminate.

In this case, clearly, $\mathcal{U}_{p(0)}$ is finite, and thus in $\mathbf{NoisyInfEx}^*$. However, $\mathbf{M}_{f(p(0))}$ on any extension of σ_s, outputs a grammar in $\text{ProgSet}(\mathbf{M}_{f(p(0))}, \sigma_s)$. Since there are at least $\text{card}(\text{ProgSet}(\mathbf{M}_{f(p(0))}, \sigma_s)) + 1$ many pairwise infinitely different languages in $\mathcal{U}_{p(0)}$ which contain $\text{content}(\sigma_s)$, it follows that $\mathbf{M}_{f(p(0))}$ does not $\mathbf{TxtFex}^*$-identify $\mathcal{U}_{p(0)}$.

From the above cases we have that $\mathbf{M}_{f(p(0))}$ does not $\mathbf{TxtFex}^*$-identify $\mathcal{U}_{p(0)} \in \mathbf{NoisyInfEx}^*$. ∎

Corollary 18 *(a) NOT* $(\exists f \in \mathcal{R})(\forall x \mid \mathcal{U}_x \in \mathbf{NoisyInfEx}^*)[\mathcal{U}_x \subseteq \mathbf{NoisyInfEx}^*(\mathbf{M}_{f(x)})]$.

(b) NOT $(\exists f \in \mathcal{R})$ $(\forall x \mid \mathcal{U}_x \in \mathbf{NoisyInfFex}^*)[\mathcal{U}_x \subseteq \mathbf{NoisyInfFex}^*(\mathbf{M}_{f(x)})]$.

Acknowledgement

A preliminary version of this paper was presented at the *Eighth International Workshop on Algorithmic Learning Theory*, Sendai, Japan. We would like to thank Frank Stephan and the referees of ALT'97 and TCS for many valuable suggestions. Research of Arun Sharma has been supported by the Australian Research Council Grant A49530274.

References

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

[B$\bar{7}$4] J. Bārzdiņš. Two theorems on the limiting synthesis of functions. In *Theory of Algorithms and Programs, vol. 1*, pages 82–88. Latvian State University, 1974. In Russian.

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

[BCJ96] G. Baliga, J. Case, and S. Jain. Synthesizing enumeration techniques for language learning. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, pages 169–180. ACM Press, 1996.

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

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

[BP73] J. Bārzdiņš and K. Podnieks. The theory of inductive inference. In *Mathematical Foundations of Computer Science*, pages 9–15. Math. Inst. of the Slovak Academy of Sciences, 1973.

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

[Cas94] J. Case. Infinitary self-reference in learning theory. *Journal of Experimental and Theoretical Artificial Intelligence*, 6(1):3–17, 1994.

[Cas96] J. Case. The power of vacillation in language learning. Technical Report LP-96-08, Logic, Philosophy and Linguistics Series of the Institute for Logic, Language and Computation, University of Amsterdam, 1996.

[CJS96] J. Case, S. Jain, and F. Stephan. Vacillatory and BC learning on noisy data. In S. Arikawa and A. Sharma, editors, *Algorithmic Learning Theory: Seventh International Workshop (ALT '96)*, volume 1160 of *Lecture Notes in Artificial Intelligence*, pages 285–298. Springer-Verlag, 1996.

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

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

[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*, pages 193–204. ACM Press, July 1996.

[Ful90] M. Fulk. Robust separations in inductive inference. *31st Annual IEEE Symposium on Foundations of Computer Science*, pages 405–410, 1990.

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

[Jan79] K. Jantke. Automatic synthesis of programs and inductive inference of functions. In *Int. Conf. Fundamentals of Computations Theory*, pages 219–225, 1979.

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

[JS97] S. Jain and A. Sharma. Characterizing language learning in terms of computable numberings. *Annals of Pure and Applied Logic*, 84(1):51–72, 1997. Special issue on Asian Logic Conference, 1993.

[Kap91] S. Kapur. *Computational Learning of Languages*. PhD thesis, Cornell University, 1991.

[KB92] S. Kapur and G. Bilardi. Language learning without overgeneralization. In *Proceedings of the Ninth Annual Symposium on Theoretical Aspects of Computer Science*, volume 577 of *Lecture Notes in Computer Science*. Springer-Verlag, 1992.

[LZK96] S. Lange, T. Zeugmann, and S. Kapur. Monotonic and dual monotonic language learning. *Theoretical Computer Science A*, 155:365–410, 1996.

[Muk92] Y. Mukouchi. Characterization of finite identification. In K. Jantke, editor, *Analogical and Inductive Inference, Proceedings of the Third International Workshop*, pages 260–267, 1992.

[Odi89] P. Odifreddi. *Classical Recursion Theory*. North-Holland, Amsterdam, 1989.

[OSW88] D. Osherson, M. Stob, and S. Weinstein. Synthesising inductive expertise. *Information and Computation*, 77:138–161, 1988.

[Shi83] T. Shinohara. Inferring unions of two pattern languages. *Bulletin of Informatics and Cybernetics*, 20:83–88., 1983.

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

[Ste95] F. Stephan. Noisy inference and oracles. In *Algorithmic Learning Theory: Sixth International Workshop (ALT '95)*, volume 997 of *Lecture Notes in Artificial Intelligence*, pages 185–200. Springer-Verlag, 1995.

[ZL95] T. Zeugmann and S. Lange. A guided tour across the boundaries of learning recursive languages. In K. Jantke and S. Lange, editors, *Algorithmic Learning for Knowledge-Based Systems*, volume 961 of *Lecture Notes in Artificial Intelligence*, pages 190–258. Springer-Verlag, 1995.

[ZLK95] T. Zeugmann, S. Lange, and S. Kapur. Characterizations of monotonic and dual monotonic language learning. *Information and Computation*, 120:155–173, 1995.