THE NATIONAL UNIVERSITY
of SINGAPORE

School of Computing
Computing 1, 13 Computing Drive, Singapore 117417

TRA7/14

On the Role of Update Constraints and Text-Types in Iterative Learning

Sanjay Jain, Timo Kötzing, Junqi Ma and Frank Stephan

July 2014

Technical Report

Foreword

This technical report contains a research paper, development or tutorial article, which has been submitted for publication in a journal or for consideration by the commissioning organization. The report represents the ideas of its author, and should not be taken as the official views of the School or the University. Any discussion of the content of the report should be sent to the author, at the address shown on the cover.

David ROSENBLUM
Dean of School

On the Role of Update Constraints and Text-Types in Iterative Learning

Sanjay Jain[1,⋆], Timo Kötzing[2], Junqi Ma[1] and Frank Stephan[1,3,⋆⋆]

[1] Department of Computer Science, National University of Singapore,
Singapore 117417, Republic of Singapore
sanjay@comp.nus.edu.sg,ma.junqi@nus.edu.sg
[2] Friedrich-Schiller University, Jena, Germany
timo.koetzing@uni-jena.de
[3] Department of Mathematics, National University of Singapore,
Singapore 119076, Republic of Singapore
fstephan@comp.nus.edu.sg

Abstract. The present work investigates the relationship of iterative learning with other learning criteria such as decisiveness, caution, reliability, non-U-shapedness, monotonicity, strong monotonicity and conservativeness. Building on the result of Case and Moelius that iterative learners can be made non-U-shaped, we show that they also can be made cautious and decisive. Furthermore, we obtain various special results with respect to one-one texts, fat texts and one-one hypothesis spaces.

1 Introduction

Iterative learning is the most common variant of learning in the limit which addresses memory constraints: the memory of the learner on past data is just its current hypothesis. Due to the padding lemma, this memory is still not void, but finitely many data can be memorised in the hypothesis. However, one subfield of the study of iterative learning considers therefore the usage of class-preserving one-one hypothesis spaces which limit this type of coding during the learning process. Other ways to limit it is to control the amount and types of updates; such constraints also aim for other natural properties of the conjectures: For example, updates have to be motivated by inconsistent data observed (syntactic conservativeness), semantic updates have to be motivated by inconsistent data observed (semantic conservativeness), updates cannot repeat semantically abandoned conjectures (decisiveness), updates cannot go from correct to incorrect hypotheses (non-U-shapedness), conjectures cannot be proper supersets of the language to be learnt (cautiousness) or conjectures have to contain all the data observed so far (consistency). There is already a quite comprehensive body of work on how iterativeness relates with various combinations of these constraints [CK10,GL04,JMZ13,JORS99,Köt09,LG02,LG03,LZ96,LZZ08], however various important questions remained unsolved. A few years ago, Case and Moelius

⋆ Supported by NUS grants C252-000-087-001 and R146-000-181-112.
⋆⋆ Supported in part by NUS grant R146-000-181-112.

[CM08b] obtained a breakthrough result by showing that iterative learners can be made non-U-shaped. The present work improves this result by showing that they can also be made decisive — this stands in contrast to the case of the usual non-iterative framework where decisiveness is a real restriction in learning [BCMSW08]. Further results complete the picture and also include the role of hypothesis spaces and text-types in iterative learning.

We completely characterise the relationship of the iterative learning criteria with the different restrictions as given in the diagramme in Figure 1. A line indicates a previously known inclusion. A gray box around criteria indicates equality of these criteria, as found in this work.

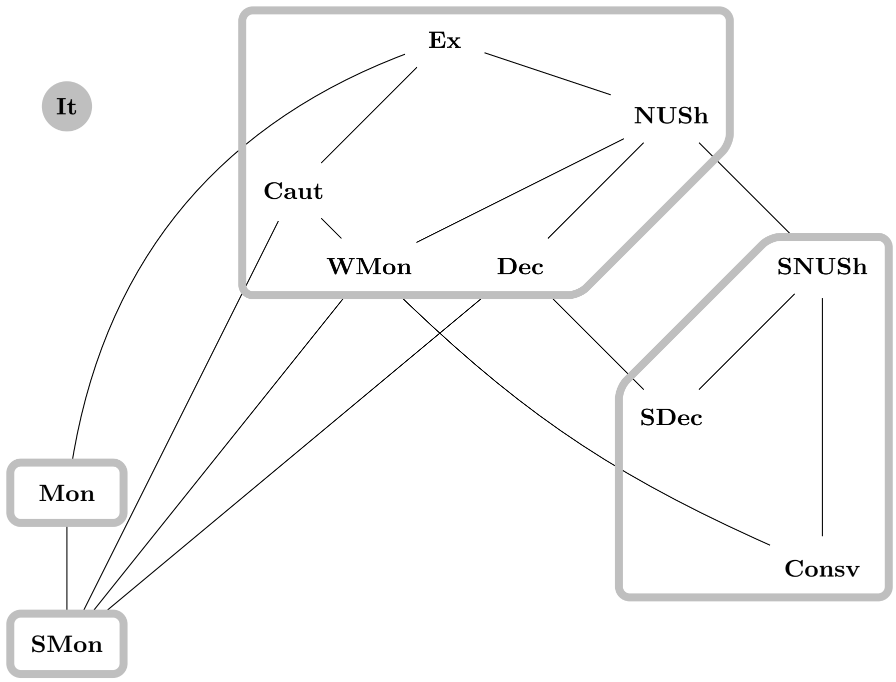


Fig. 1. Relation of criteria combined with iterative learning.

The learning criteria investigated in the present work are quite natural. Conservativeness, consistency, cautiousness and decisiveness are natural constraints studied for a long time [Ang80,OSW86]; these criteria require that conjectures contain the data observed (consistency) or that mind changes are based on evidence that the prior hypothesis is incorrect (conservativeness); a lot of work has been undertaken using the assumption that learners are both, consistent and conservative. Monotonicity constraints play an important role in various fields like monotonic versus non-monotonic logic and this is reflected in inductive inference by considering the additional requirement that new hypotheses should be at least as general as the previous ones [Jan91,LZ93]. The fundamental notion of iterative learning is one of the first memory-constraints to be investigated in inductive inference and has been widely studied [LG02,LG03,LZ96,OSW86]; the beauty of this criterion is that the memory limitation comes rather indirectly, as for finitely many steps the memory can be enhanced by padding; after that, however, the learner has to converge and to ignore new data unless it gives enough evidence to undertake a mind change. Osherson, Stob and Weinstein [OSW82] formalised decisiveness as a notion where a learner never semantically returns to an abandoned hypothesis; they left it as an open problem whether the notion of decisiveness is restrictive; it took about two decades until the problem was solved [BCMSW08]. The search for this solution and also the parallels to developmental psychology motivated to study the related notion of non-U-shapedness where a non-U-shaped learner never abandons a correct hypothesis for an incorrect one and later (in a U-shaped way) returns to a correct hypothesis. The study of this field turned out to be quite fruitful and productive and we also consider decisive and non-U-shaped learning and its variants in this paper.

Taking this into account, we believe that the criteria investigated are natural and deserve to be studied; the restrictions on texts which we investigated are motivated from the fact that in the case of memory limitations (like enforced by iterativeness), the learners cannot keep track of which information has been presented before and therefore certain properties of the text (like every datum appearing exactly once or every datum appearing infinitely often) can be exploited by the learner during the learning process. In some cases these exploitations only matter when the restrictions on the hypothesis space make the iterativeness-constraint stricter, as they might rule out padding. Such a restriction is quite natural, as padding is a way to permit finite calculations to go into the update process and thereby bypass the basic idea behind the notion of iterativeness; this is reflected in the finding that the relations between the learning criteria differ for iterative learning in general and iterative learning using a class-preserving one-one hypothesis space.

2 Mathematical Preliminaries

Unintroduced notation follows the textbook of Rogers [Rog67] on recursion theory. The set of natural numbers is denoted by $\mathbb{N} = \{0, 1, 2, \ldots\}$. The symbols $\subseteq$, $\subset$, $\supseteq$, $\supset$ respectively denote the subset, proper subset, superset and proper superset relation between sets. The symbol $\emptyset$ denotes both the empty set and the empty sequence.

With dom and range we denote, respectively, domain and range of a given function. We sometimes denote a partial function f of $n > 0$ arguments $x_1, \ldots, x_n$ in lambda notation (as in Lisp) as $\lambda x_1, \ldots, x_n . f(x_1, \ldots, x_n)$. For example, with $c \in \mathbb{N}$, $\lambda x . c$ is the constantly c function of one argument.

We let $\langle x, y \rangle = \frac{(x+y)(x+y+1)}{2} + x$ be Cantor's Pairing function which is an invertible, order-preserving function from $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$. Whenever we consider tuples of natural numbers as input to a function, it is understood that the general coding function $\langle \cdot, \cdot \rangle$ is used to code the tuples into a single natural number. We similarly fix a coding for finite sets and sequences, so that we can use those as input as well.

If a function f is not defined for some argument x, then we denote this fact by $f(x)\uparrow$ and we say that f on x *diverges*; the opposite is denoted by $f(x)\downarrow$ and we say that f on x *converges*. If f on x converges to p, then we denote this fact by $f(x)\downarrow = p$.

$\mathcal{P}$ and $\mathcal{R}$ denote, respectively, the set of all partial recursive and the set of all recursive functions (mapping $\mathbb{N} \to \mathbb{N}$). We let φ be any fixed acceptable numbering for $\mathcal{P}$ (an acceptable numbering could, for example, be based on a natural programming language such as C or Java). Further, we let φ_p denote the partial-recursive function computed by the φ-program with code number p. A set $L \subseteq \mathbb{N}$ is *recursively enumerable (r.e.)* iff it is the domain of a partial recursive function. We let $\mathcal{E}$ denote the set of all r.e. sets. We let W be the mapping such that $\forall e : W_e = \mathrm{dom}(\varphi_e)$. W is, then, a mapping from $\mathbb{N}$ *onto* $\mathcal{E}$. We say that e is an index, or program, (in W) for W_e. Let $W_{e,s}$ denote W_e enumerated in s steps in some uniform way to enumerate all the W_e's. We let pad be a 1–1 padding function such that for all e and finite sets D, $W_{\mathrm{pad}(e,D)} = W_e$.

The special symbol ? is used as a possible hypothesis (meaning "no change of hypothesis"). The symbol # stands for a pause, that is, for "no new input data in the text". For each (possibly infinite) sequence q with its range contained in $\mathbb{N} \cup \{\#\}$, let $\mathrm{content}(q) = (\mathrm{range}(q) \setminus \{\#\})$. By using an appropriate coding, we assume that ? and # can be handled by recursive functions.

For any function f and all i, we use $f[i]$ to denote the sequence $f(0), \ldots, f(i-1)$ (the empty sequence if $i = 0$ and undefined, if one of these values is undefined).

3 Learning Criteria

In this section we formally introduce our setting of learning in the limit and associated learning criteria. We follow [Köt09] in its "building-blocks" approach for defining learning criteria.

A *learner* is a partial function from $\mathbb{N}$ to $\mathbb{N} \cup \{?\}$. A *language* is a r.e. set $L \subseteq \mathbb{N}$. Any total function $T : \mathbb{N} \to \mathbb{N} \cup \{\#\}$ is called a *text*. For any given language L, a *text for* L is a text T such that $\mathrm{content}(T) = L$. Initial parts of this kind of text is what learners usually get as information. We let σ and τ range over initial segments of texts. Concatenation of two initial segments σ and τ is denoted by $\sigma \diamond \tau$. For a given set of texts F, we let $\mathbf{Txt}^F(L)$ denote the set of all texts in F for L.

An *interaction operator* is an operator β taking as arguments a function M (the learner) and a text T, and that outputs a function p. We call p the *learning sequence* (or *sequence of hypotheses*) of M given T. Intuitively, β defines how a learner can interact with a given text to produce a sequence of conjectures.

We define the sequence generating operators $\mathbf{G}$ and $\mathbf{It}$ (corresponding to the learning criteria discussed in the introduction) as follows. For all learners M, texts T and all i,

$$\mathbf{G}(M,T)(i) = M(T[i]);$$
$$\mathbf{It}(M,T)(i) = \begin{cases} M(\emptyset), & \text{if } i = 0; \\ M(\mathbf{It}(M,T)(i-1), T(i-1)), & \text{otherwise;} \end{cases}$$

where $M(\emptyset)$ denotes the *initial conjecture* made by M. Thus, in iterative learning, the learner has access to the previous conjecture, but not to all previous data as in $\mathbf{G}$-learning. With any iterative learner M we associate a learner M^* such that

$$M^*(\emptyset) = M(\emptyset) \text{ and}$$
$$\forall \sigma, x : M^*(\sigma \diamond x) = M(M^*(\sigma), x).$$

Intuitively, M^* on a sequence σ returns the hypothesis which M makes after being fed the sequence σ in order. Note that, for all texts T, $\mathbf{G}(M^*, T) = \mathbf{It}(M, T)$. We let $M(T)$ (respectively $M^*(T)$) denote $\lim_{n\to\infty} M(T[n])$ (respectively, $\lim_{n\to\infty} M^*(T[n])$) if it exists.

Successful learning requires the learner to observe certain restrictions, for example convergence to a correct index. These restrictions are formalised in our next definition.

A *learning restriction* is a predicate δ on a learning sequence and a text. We give the important example of explanatory learning (**Ex**, [Gol67]) and that of vacillatory learning (**Fex**, [CL82,OW82,Cas99]) defined such that, for all sequences of hypotheses p and all texts T,

$$\begin{aligned}
\mathbf{Ex}(p, T) &\Leftrightarrow [\exists n_0 \forall n \geq n_0 : p(n) = p(n_0) \wedge W_{p(n_0)} = \mathrm{content}(T)];\\
\mathbf{Fex}(p, T) &\Leftrightarrow [\exists n_0 \exists \text{ finite } D \subset \mathbb{N}\\
&\qquad\qquad \forall n \geq n_0 : p(n) \in D \wedge \forall e \in D : W_e = \mathrm{content}(T)].
\end{aligned}$$

Furthemore, we formally define the restrictions discussed in Section 1 in Figure 2. We combine

$$\begin{aligned}
\mathbf{Consv}(p, T) &\Leftrightarrow [\forall i : \mathrm{content}(T[i+1]) \subseteq W_{p(i)} \Rightarrow p(i) = p(i+1)];\\
\mathbf{Caut}(p, T) &\Leftrightarrow [\forall i, j : W_{p(i)} \subset W_{p(j)} \Rightarrow i < j];\\
\mathbf{NUSh}(p, T) &\Leftrightarrow [\forall i, j, k : i \leq j \leq k \ \wedge\ W_{p(i)} = W_{p(k)} = \mathrm{content}(T) \Rightarrow W_{p(j)} = W_{p(i)}];\\
\mathbf{Dec}(p, T) &\Leftrightarrow [\forall i, j, k : i \leq j \leq k \ \wedge\ W_{p(i)} = W_{p(k)} \Rightarrow W_{p(j)} = W_{p(i)}];\\
\mathbf{SNUSh}(p, T) &\Leftrightarrow [\forall i, j, k : i \leq j \leq k \ \wedge\ W_{p(i)} = W_{p(k)} = \mathrm{content}(T) \Rightarrow p(j) = p(i)];\\
\mathbf{SDec}(p, T) &\Leftrightarrow [\forall i, j, k : i \leq j \leq k \ \wedge\ W_{p(i)} = W_{p(k)} \Rightarrow p(j) = p(i)];\\
\mathbf{SMon}(p, T) &\Leftrightarrow [\forall i, j : i < j \Rightarrow W_{p(i)} \subseteq W_{p(j)}];\\
\mathbf{Mon}(p, T) &\Leftrightarrow [\forall i, j : i < j \Rightarrow W_{p(i)} \cap \mathrm{content}(T) \subseteq W_{p(j)} \cap \mathrm{content}(T)];\\
\mathbf{WMon}(p, T) &\Leftrightarrow [\forall i, j : i < j \wedge \mathrm{content}(T[j]) \subseteq W_{p(i)} \Rightarrow W_{p(i)} \subseteq W_{p(j)}].
\end{aligned}$$

Fig. 2. Definitions of learning restrictions.

any two sequence acceptance criteria δ and δ' by intersecting them; we denote this by juxtaposition (for example, all the restrictions given in Figure 2 are meant to be always used together with **Ex**).

For any set of texts F, interaction operator β and any (combination of) learning restrictions δ, $\mathbf{Txt}^F\beta\delta$ is a *learning criterion*. A learner M $\mathbf{Txt}^F\beta\delta$*-learns* all languages in the class

$$\mathbf{Txt}^F\beta\delta(M) = \{L \in \mathcal{E} \mid \forall T \in \mathbf{Txt}(L) \cap F : \delta(\beta(M, T), T)\}$$

and we use $\mathbf{Txt}\beta\delta$ to denote the set of all $\mathbf{Txt}\beta\delta$-learnable classes (learnable by some learner). Note that we omit the superscript F whenever F is the set of all texts.

In some cases, we consider learning using an explicitly given particular hypothesis space $(H_e)_{e\in\mathbb{N}}$ instead of the usual acceptable numbering $(W_e)_{e\in\mathbb{N}}$. For this, one replaces W_e by H_e in the respective definitions of learning as above.

4 Plain-Text Learning

In this section we first show that, for iterative learning, the convergence restrictions **Ex** and **Fex** allow for learning the same sets of languages. After that we give the necessary theorems establishing the diagramme given in Figure 1.

Theorem 1. **TxtItFex** = **TxtItEx**.

Proof. Clearly, **TxtItEx** $\subseteq$ **TxtItFex**. Suppose a learner M **TxtItFex**-learning a class $\mathcal{L}$ is given. Now we construct a learner N as follows. N keeps track of all the past conjectures of M and does not change mind if M changes its mind to a conjecture made in the past. The initial output of N is $\mathrm{pad}(i, \emptyset)$, where i is the initial conjecture of M. For the update of N with current hypothesis $\mathrm{pad}(p, D)$ and input x, if $M(p, x) \in D \cup \{p\}$, then let $N(\mathrm{pad}(p, D), x) = \mathrm{pad}(p, D)$, else let $N(\mathrm{pad}(p, D), x) = \mathrm{pad}(M(p, x), D \cup \{p\})$.

Now, for the verification of the learner, consider a text $T = x_0 \diamond x_1 \diamond x_2 \diamond \ldots$ for a language $L \in \mathcal{L}$. We will below define another text $T' = x_0 \diamond \tau_0 \diamond x_1 \diamond \tau_1 \ldots$ such that for all n,

(E1) $N^*(x_0 \diamond x_1 \diamond \ldots x_{n-1}) = (M^*(x_0 \diamond \tau_0 \diamond x_1 \diamond \tau_1 \ldots x_{n-1} \diamond \tau_{n-1}), \{N^*(\emptyset), N^*(x_0), N^*(x_0 \diamond x_1), \ldots, N^*(x_0 \diamond x_1 \diamond \ldots \diamond x_{n-1})\} - \{N^*(x_0 \diamond x_1 \diamond \ldots \diamond x_{n-1})\})$

where for $n = 0$, we take the input sequences for M and N as empty in the above equation.

Note that (E1) is void for $n = 0$. So suppose we have defined $\tau_0, \tau_1, \ldots, \tau_m$ and (E1) holds for all $n \leq m$. Then, consider $n = m + 1$. Suppose $N^*(x_0 \diamond x_1 \diamond \ldots \diamond x_{m-1}) = (p, D)$. If $M(p, x_m) \notin D \cup \{p\}$, then let $\tau_m = \emptyset$. If $M(p, x_m) \in D \cup \{p\}$, then let $m' < m$ be least such that $M(p, x_m) = N^*(x_0 \diamond x_1 \diamond \ldots \diamond x_{m'})$; then, let $\tau_m = x_{m'+1} \diamond \tau_{m'+1} \diamond x_{m'+2} \diamond \tau_{m'+2} \diamond \ldots \diamond x_{m-1} \diamond \tau_{m-1}$, where if $m' = m - 1$, then $\tau_m = \emptyset$. It is easy to verify that (E1) holds. Furthermore, note that τ_m only consists of elements from $x_0, x_1, \ldots, x_{m-1}$.

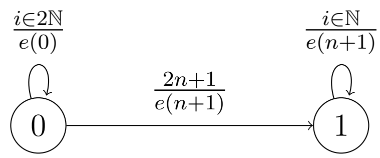
Now consider the sequence formed using the first component of the sequence of outputs of N on $T = x_0 \diamond x_1 \diamond \ldots$; this sequence is a subsequence of the outputs of M on $T' = x_0 \diamond \tau_0 \diamond x_1 \diamond \tau_1 \ldots$ and thus, due to padding, the sequence of hypotheses of N on T has a similar learning behaviour as the one of M on T'. Note that T' is also a text for $\mathrm{content}(T) = L$. Also note that the second component in the conjectures of N on T is a finite subset of conjectures of M on T' and this sequence is monotonically non-decreasing on T. Furthermore, clearly, if $N^*(\sigma \diamond x) \neq N^*(\sigma)$, then $N^*(\sigma \diamond x \diamond \tau) \neq N^*(\sigma)$. Thus, N **TxtItEx** learns every language **TxtItFex**-learnt by M. $\square$

Next we give separating theorems for monotone learning and first show that there is a class which can be learnt iteratively by a learner which is strongly decisive, conservative, monotone and cautious while on the other hand, there is no learner which, even non-iteratively, learns the same class strongly monotonically.

Theorem 2. **TxtItSDecConsvMonCautEx** $\not\subseteq$ **TxtGSMonEx**.

Proof. Let $L_0 = \{0, 2, 4, \ldots\}$ and for all n, $L_{n+1} = \{2m \mid m \leq n\} \cup \{2n + 1\}$. Let $\mathcal{L} = \{L_n : n \in \mathbb{N}\}$. Let e be a recursive function computing an r.e. index for L_n: $W_{e(n)} = L_n$. Let $M \in \mathcal{P}$

be the iterative learner which memorises a single state in its conjecture (using padding) and has the following state transition diagramme (an edge labeled $\frac{x}{e}$ means that the edge indicates a state transition on input x with conjecture output e).



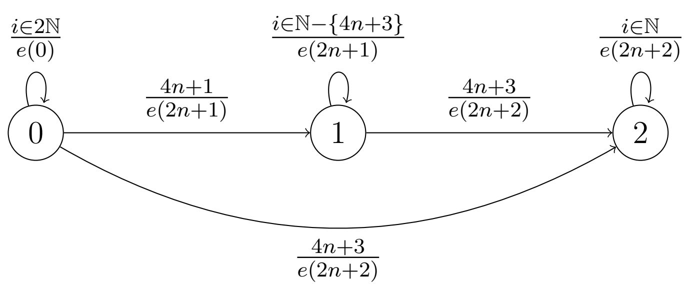
Clearly, M is a **TxtItSDecConsvMonCautEx**-learner for $\mathcal{L}$. The class $\mathcal{L}$ is not strongly monotonically learnable as a learner for it eventually conjecture L_0 after seeing finitely many even numbers. Then, for n larger than all previous data seen, the learner must change its conjecture to L_{n+1}, if the rest of the data is for L_{n+1}, this mind change is not strongly monotonic. □

Note that one can modify this protocol such that M only memorises the state; however, M then abstains from repeating correct conjectures and one has to modify the learnability criterion such that outputting a special symbol for repeating the last (correct) conjecture is allowed. The next result shows that there is a class of languages which can be learnt by an iterative learner which is strongly decisive, conservative and cautious; on the other hand, there is no learner, even non-iterative one, that learns the class monotonically.

Theorem 3. **TxtItSDecConsvCautEx** $\not\subseteq$ **TxtGMonEx**.

Proof. We consider $L_0 = \{0, 2, 4, \ldots\}$ and, for all n, $L_{2n+1} = \{2m \mid m \leq n\} \cup \{4n+1\}$ and $L_{2n+2} = \{2m \mid m \leq n+1\} \cup \{4n+1, 4n+3\}$. We let $\mathcal{L} = \{L_n \mid n \in \mathbb{N}\}$.

Let e be a recursive function such that, for all n, $W_{e(n)} = L_n$. Let $M \in \mathcal{P}$ be the iterative learner using state transitions as given by the following diagramme.



Clearly, M fulfills all the desired requirements for **TxtItSDecConsvCautEx**-learning $\mathcal{L}$. One can show that every learner of $\mathcal{L}$ outputs on some text for some L_{2n+2} hypotheses for L_0, L_{2n+1} and L_{2n+2} (in that order, with possibly other hypotheses in between) and is therefore not learning monotonically. □

The next result shows that there is a class of languages which is simultaneously iteratively, monotonically, decisively, weakly monotonically and cautiously learnable, but not iteratively strongly non-U-shapedly learnable.

Theorem 4. **TxtItMonDecWMonCautEx** $\not\subseteq$ **TxtItSNUShEx**.

Proof. Case and Kötzing [CK10] provided a class which separates **NUSh** from **SNUSh** and also shows this more general theorem. The result furthermore also follows from Theorem 27 below which does not only diagonalise against conservative learners but also against learners which never update a correct hypothesis. ☐

The next result shows that there is an iteratively and strongly monotonically learnable class which does not have any iterative learner which is strongly non-U-shaped, that is, which never revises a correct hypothesis. The proof uses the notion of a join which is defined as $A \oplus B = \{2x : x \in A\} \cup \{2x + 1 : x \in B\}$.

Theorem 5. **TxtItSMonEx** $\not\subseteq$ **TxtItSNUShEx**.

Proof. Let $M_0, M_1, \ldots$ denote a recursive listing of all partial recursive iterative learning machines. Consider a class $\mathcal{L}$ consisting of the following sets for each $e \in \mathbb{N}$ (where $F(\cdot)$, $G(\cdot)$ are recursively enumerable sets in the parameters described later):

– $\{2e\} \oplus F(e)$
– $\{2e, 2d + 1\} \oplus G(e, d)$
– $\{2e, 2d + 1\} \oplus \mathbb{N}$

where,

(a) If there exists an s such that $M_e^*(4e \diamond 1 \diamond \# \diamond 3 \diamond \# \diamond 5 \diamond \# \ldots \diamond 2s + 1) = M_e^*(4e \diamond 1 \diamond \# \diamond 3 \diamond \# \diamond 5 \diamond \# \ldots \diamond 2s + 1 \diamond \# \diamond 2s' + 1)$, for all $s' > s$, then $F(e) = \{0, 1, 2, \ldots, s\}$, else $F(e) = \mathbb{N}$.
(b) If $F(e) = \mathbb{N}$ or $\max(F(e)) > d$, then $G(e, d) = \mathbb{N}$. Otherwise, if there exists a $k > d$ such that $M_e^*(4e \diamond 1 \diamond \# \diamond 3 \diamond \# \diamond 5 \diamond \# \ldots \diamond 2\max(F(e)) + 1 \diamond \# \diamond 4d + 2 \diamond \#^r) = M_e^*(4e \diamond 1 \diamond \# \diamond 3 \diamond \# \diamond 5 \diamond \# \diamond \ldots \diamond 2max(F(e)) + 1 \diamond \# \diamond 4d + 2 \diamond \#^r \diamond \#) \neq M_e^*(4e \diamond 1 \diamond \# \diamond 3 \diamond \# \diamond 5 \diamond \# \ldots \diamond 2max(F(e)) + 1 \diamond \# \diamond 4d + 2 \diamond \#^r \diamond \# \diamond 2k + 1)$ then $G(e, d) = F(e) \cup \{k\}$ for first such k found in some algorithmic search, else $G(e, d) = F(e)$.

Now, the above class is **TxtItSMonEx** learnable, as the learner can remember seeing $4e, 4d + 2$ in the input text, if any:

- Having seen only $4e$, the learner outputs a grammar for $\{2e\} \oplus F(e)$;
- Having seen $4e, 4d + 2$, the learner outputs a grammar for $\{2e, 2d + 1\} \oplus G(e, d)$ until it sees, (after having seen $4e, 4d + 2$), two more odd elements bigger than $2d$ in the input, at which point the learner switches to outputting a grammar for $\{2e, 2d + 1\} \oplus \mathbb{N}$.

It is easy to verify that the above learner will **TxtItSMon** learn $\mathcal{L}$.

Now we show that $\mathcal{L}$ is not **TxtItSNUShEx**-learnable. Suppose by way of contradiction that M_e **TxtItSNUShEx**-learns $\mathcal{L}$. Then the following statements hold:

– There exists an s as described in the definition of $F(e)$ above and thus $F(e)$ is finite, as otherwise M_e does not learn $2e \oplus F(e) = 2e \oplus \mathbb{N}$;

– For $d > \max(F(e))$, there exists a $k > d$ as described in the definition of $G(e, d)$, as otherwise M_e does not learn at least one of $\{2e, 2d+1\} \oplus G(e, d)$ and $\{2e, 2d+1\} \oplus \mathbb{N}$;
– Now the learner M_e has two different hypotheses on the segments $(4e\diamond 1\diamond\#\diamond 3\diamond\#\diamond\ldots\diamond 2F(e)+1\diamond\#\diamond 2k+1\diamond\#\diamond 4d+2\diamond\#^r)$ and $(4e\diamond 1\diamond\#\diamond 3\diamond\#\diamond\ldots\diamond 2F(e)+1\diamond\#\diamond 2k+1\diamond\#\diamond 4d+2\diamond\#^r\diamond 2k+1)$ and first of them must be correct hypothesis for $\{2e, 2d+1\}\oplus G(e, d)$, as otherwise the learner M_e does not learn it from the text — $4e\diamond 1\diamond\#\diamond 3\diamond\#\diamond\ldots\diamond 2F(e)+1\diamond\#\diamond 2k+1\diamond\#\diamond 4d+2\diamond\#^r\diamond\#^\infty$ — see part (b) in the definition of $G(e, d)$, whereas second is a mind change, after the correct hypothesis by M_e on $\{2e, 2d+1\} \oplus G(e, d)$.

Thus, M_e does not **TxtItSNUShEx**-learn $\mathcal{L}$. □

For our following proofs we will require the notion of a *canny* learner [CM08b].

Definition 6 (Case and Moelius [CM08b]). For all iterative learners M, we say that M is *canny* iff

1. M never outputs ?,
2. for all e, $M(e, \#) = e$ and
3. for all x, τ and σ, if $M^*(\sigma \diamond x) \neq M^*(\sigma)$ then $M^*(\sigma \diamond x \diamond \tau \diamond x) = M^*(\sigma \diamond x \diamond \tau)$.

Case and Moelius [CM08b] showed that, for **TxtItEx**-learning, learners can be assumed to be canny.

Lemma 7 (Case and Moelius [CM08b]). *For all* $\mathcal{L} \in$ **TxtItEx** *there exists canny iterative learner* M *such that* $\mathcal{L} \subseteq$ **TxtItEx**(M).

The term "sink-locking" means that on any text for a language to be learnt the learner converges to a *sink*, a correct hypothesis which is not abandoned on any continuation of the text. The following result does not only hold for the case where all texts are allowed but also for the case where only fat texts are allowed (see Section 5). As both proofs are similar, only the standard case of all texts is given.

Theorem 8. *Let* $\mathcal{L}$ *be sink-lockingly* **TxtItEx**-*learnable. Then* $\mathcal{L}$ *is cautiously, conservatively, strongly decisively and weakly monotonically* **TxtItEx**-*learnable.*

Proof. Let M be a sink-locking **TxtItEx**-learner for $\mathcal{L}$. Using the S-m-n Theorem, we let $f \in \mathcal{R}$ be such that

$$\forall e : W_{f(e)} = \{x \in W_e \mid M(e, x) = e\}.$$

Let N be such that $N^*(\sigma) = f(M^*(\sigma))$ for all sequences σ. From M sink-locking we now immediately get that N is a conservative and weakly monotone iterative learner for $\mathcal{L}$. Again with the S-m-n Theorem we let $g \in \mathcal{R}$ be such that, for all σ,

$$W_{g(\sigma)} = \begin{cases} \emptyset, & \text{if } \sigma = \emptyset; \\ \{x \leq 2|\sigma|\} \setminus \{\sigma(0)\}, & \text{if } \sigma \neq \emptyset \text{ and } N^*(\sigma) \neq N^*(\sigma \diamond \#); \\ W_{N^*(\sigma)} \cup \text{content}(\sigma), & \text{otherwise.} \end{cases}$$

We let O be an iterative learner with initial conjecture $g(\emptyset)$ and, given a current conjecture, $g(\sigma)$ and a new datum x,

$$O(g(\sigma), x) = \begin{cases} g(\emptyset) & \text{if } \sigma = \emptyset \text{ and } x = \#; \\ g(\sigma), & \text{if } N^*(\sigma) = N^*(\sigma \diamond x); \\ g(\sigma \diamond x), & \text{otherwise.} \end{cases}$$

As N is conservative, it is straightforward to see that O **TxtItEx**-learns $\mathcal{L}$. Next we show that O is strongly decisive. To that end we observe that, for each hypothesis $g(\sigma)$ made by O, we have $\sigma(0) \neq \#$, so that the hypothesis given in one case of g is (semantically) distinct from any hypothesis given in a different case. The same holds trivially for hypothesis within the second case.

Let now σ, τ, x be given such that $N^*(\sigma) = N^*(\sigma \diamond \#)$, $N^*(\sigma \diamond x \diamond \tau) = N^*(\sigma \diamond x \diamond \tau \diamond \#)$ and $x \neq \#$, so that $g(\sigma)$ and $g(\sigma \diamond x \diamond \tau)$ might be conjectures output by O with g being in the third case both times. We have $x \in W_{g(\sigma \diamond x \diamond \tau)} \setminus W_{g(\sigma)}$ (x is in the first set due to the construction of g and not in the second set due to N being conservative). Thus, O is strongly decisive. □

The previous theorem gives us the following immediate corollary which states that a class is iteratively strongly decisive learnable from text iff it is iteratively conservatively learnable from text iff it is iteratively strongly non-U-shaped learnable from text.

Corollary 9. *We have that*

$$\mathbf{TxtItSDecEx} = \mathbf{TxtItConsvEx} = \mathbf{TxtItSNUShEx}.$$

Proof. We have that strongly decisive or conservative (iterative) learnability trivially implies strongly non-U-shaped learnability. Using Theorem 8 it remains to show that strongly non-U-shaped learnability implies sink-locking learnability. But this is trivial, as the learner can never converge to a correct conjecture that might possibly be abandoned on the given language, as this would contradict strong non-U-shapedness. □

Case and Moelius [CM08b] showed that **TxtItNUShEx** = **TxtItEx**; we finally show that this proof can be extended to also cover decisiveness, weak monotonicity and caution.

Theorem 10. *We have that*

$$\mathbf{TxtItEx} = \mathbf{TxtItDecEx} = \mathbf{TxtItWMonEx} = \mathbf{TxtItCautEx}.$$

Proof. Suppose M is a canny iterative learner which learns a class $\mathcal{L}$. Below we will construct an iterative learner N which is weakly monotonic and learns $\mathcal{L}$. Let

$$\begin{aligned} C_M(\sigma) &= \{x \in \mathbb{N} \cup \{\#\} : M^*(\sigma \diamond x)\downarrow = M^*(\sigma)\downarrow\}; \\ B_M(\sigma) &= \{x \in \mathbb{N} \cup \{\#\} : M^*(\sigma \diamond x)\downarrow \neq M^*(\sigma)\downarrow\}; \\ B_M^{\cap}(\sigma) &= \bigcap_{0 \leq i \leq |\sigma|} B_M(\sigma[i]); \\ CB_M(\sigma) &= \bigcup_{0 \leq i < |\sigma|} C_M(\sigma[i]) \cap B_M(\sigma). \end{aligned}$$

Let P be such that for all σ and m and $x \in \mathbb{N} \cup \{\#\}$, $P(\sigma, m, x)$ iff (i) $x \neq \#$ and (ii) $(\exists w)[M^*(\sigma \diamond w)$ converges in x steps, $W_{M^*(\sigma)}$ enumerates w in x steps, $w \in CB_M(\sigma)$ and $m < w \leq x]$.

Let N be such that $N(\emptyset) = f(\emptyset, 0, \emptyset)$, and for all inputs x, and previous conjecture $f(\sigma, m, \alpha)$, N outputs as follows:

$$\begin{cases} \uparrow, & \text{(i) if } M^*(\tau)\uparrow \text{ for some } \tau \in \{\sigma, \sigma \diamond \alpha, \sigma \diamond x, \sigma \diamond \alpha \diamond x\}; \\ f(\sigma \diamond \alpha \diamond x, 0, \emptyset), & \text{(ii) if } \neg \text{ (i) and } (x \in B^{\cap}_M(\sigma) \text{ or } (x \in CB_M(\sigma) \text{ and } x > m)); \\ f(\sigma, m, \alpha \diamond x), & \text{(iii) if } \neg \text{ ((i) or (ii)) and} \\ & x \in CB_M(\sigma \diamond \alpha) \\ f(\sigma, x, \emptyset), & \text{(iv) if } \neg \text{ ((i) or (ii)) and} \\ & x \in C_M(\sigma \diamond \alpha) \text{ and } P(\sigma, m, x) \text{ and } \alpha = \emptyset; \\ f(\sigma \diamond \alpha \diamond x, 0, \emptyset), & \text{(v) if } \neg \text{ ((i) or (ii)) and} \\ & x \in C_M(\sigma \diamond \alpha) \text{ and } P(\sigma, m, x) \text{ and } \alpha \neq \emptyset; \\ f(\sigma, m, \alpha), & \text{(vi) if } \neg \text{ ((i) or (ii)) and} \\ & x \in C_M(\sigma \diamond \alpha) \text{ and } \neg P(\sigma, m, x). \end{cases}$$

Here $W_{f(\sigma,m,\alpha)}$ is defined as follows.

1. Enumerate content(σ)
   In the following, if the needed $M^*(\cdot)$ (to compute various parameters), is not defined, then do not enumerate any more.
2. Go to stage 0.
   Stage s:
      Let $A_s = \text{content}(\sigma) \cup W_{M^*(\sigma),s}$
   (a) If there exists an $x \in A_s$ such that $x \in B^{\cap}_M(\sigma)$, then no more elements are enumerated.
   (b) If there exists an $x \in A_s$ such that $x > m$, and $[x \in CB_M(\sigma)$ or $P(\sigma, m, x)]$, then:
      If for all τ with content(τ) $\subseteq A_s$ and $|\tau| \leq |A_s| + 1$, τ not containing $\#$ and τ starting with a y in $CB_M(\sigma)$: $A_s \subseteq W_{f(\sigma\diamond\tau,0,\emptyset)}$,
      then enumerate A_s and go to stage $s+1$;
      otherwise, no more elements are enumerated.
      (basically, this is testing if x satisfies clauses ii, iv or v in the defn of M)
   (c) If both (a) and (b) fail, then enumerate A_s, and go to stage $s+1$.

   End stage s

It can be easily shown by induction on the length of ρ, that for all input ρ, if $N^*(\rho) = f(\sigma, m, \alpha)$, then $M^*(\rho) = M^*(\sigma \diamond \alpha)$.

Now, for finite languages L iteratively learnt by M, if content(σ) $\subseteq L$ and $L \cap B^{\cap}_M(\sigma) = \emptyset$, then $W_{M^*(\sigma)} = L$. To see this note that if we construct a sequence τ from σ, by inserting elements of $L -$ content(σ) after the initial segment σ' of σ such that $x \in C_M(\sigma')$, then $M^*(\sigma) = M^*(\tau)$, and content(τ) $= L$; thus, $M^*(\sigma) = M^*(\sigma\#^\infty) = M^*(\tau\#^\infty)$, which must be a grammar for L.

Thus for such σ, for content$(\alpha) \subseteq L$, using the fact that M is canny and using reverse induction on the number of mind changes made by M on σ (which is bounded by card(L) due to M being canny), it is easy to verify that $W_{f(\sigma,m,\alpha)}$ would be L.

Given an infinite languages $L \in \mathcal{L}$ and a text T for L, consider the output $f(\sigma_n, m_n, \alpha_n)$ of $N^*(T[n])$. As $M^*(T)$ converges, $\sigma = \lim_{n\to\infty} \sigma_n$ and $\lim_{n\to\infty} \alpha_n$ would converge. For this paragraph fix this σ and α. If $\alpha \neq \emptyset$, then clearly $m = \lim_{n\to\infty} m_n$ also converges, and as $B_M^{\cap}(\sigma) \cap L = \emptyset$, we also have $W_{M^*(\sigma)} = L$. If $\alpha = \emptyset$, then as $M^*(T) = M^*(\sigma)$, we have that $W_{M^*(\sigma)} = L$ and all but finitely many of the elements of L do not belong to $B_M(\sigma)$. Thus, in this case also $m = \lim_{n\to\infty} m_n$ converges. In both cases, m bounds all the elements of L which are in $B_M(\sigma)$. Thus, $f(\sigma, m, \alpha)$ would be a grammar for L.

Now we show that N is weakly monotonic. Note that, for all σ, α, m, $W_{f(\sigma,m,\alpha)} \subseteq \text{content}(\sigma) \cup W_{M^*(\sigma)}$.

Also, note that $W_{f(\sigma,m,\alpha)} \subseteq W_{f(\sigma,m+1,\alpha')}$ for all $m, \alpha, \sigma, \alpha'$ — (P1).

Now suppose N on input $\rho \diamond x$ and previous conjecture (on input ρ) being $f(\sigma, m, \alpha)$ outputs $f(\sigma \diamond \alpha \diamond x, 0, \emptyset)$. This implies that, $x \in B_M^{\cap}(\sigma)$ or $x > m$ and $(CB_M(\sigma)$ or $P(\sigma, m, x))$ hold.

Case 1: content$(\alpha \diamond x)$ is not contained in $W_{f(\sigma,m,\alpha)}$.

In this case clearly content$(\rho \diamond x) \supseteq$ content$(\sigma \diamond \alpha \diamond x)$ and thus, content$(\rho \diamond x)$ is not contained in $W_{f(\sigma,m,\alpha)}$, so mind change is safe.

Case 2: content$(\alpha \diamond x)$ is contained in $W_{f(\sigma,m,\alpha)}$ and thus in content$(\sigma) \cup W_{M^*(\sigma)}$.

Let s be least such that content$(\alpha \diamond x)$ is contained in A_s as in stage s. Then, the definition of $W_{f(\sigma,m,\alpha)}$ ensures that $W_{f(\sigma,m,\alpha)}$ enumerates $A_t, t \geq s$, only if A_t is contained in $W_{f(\sigma\diamond\alpha\diamond x,0,\emptyset)}$ (note that the case of $A_t =$ content(σ), already satisfies $A_t \subseteq W_{f(\sigma\diamond\alpha\diamond x,0,\emptyset)}$).

It follows from the above analysis that either the new input is not contained in the previous conjecture of N, or the previous conjecture is contained in the new conjecture. Thus, N is weakly monotonic.

It follows from the above construction that N is also decisive and cautious. To see this, note that whenever mind change of N falls in Case 1 above, all future conjectures of N (beyond input $\rho \diamond x$) contain content$(\alpha \diamond x)$; thus, N never returns to the conjecture $W_{f(\sigma,m,\alpha)}$, which does not contain content$(\alpha \diamond x)$. On the other hand, the mind changes due to Case 2 or mind changes due to N outputting $f(\sigma, m', \alpha')$ after outputting $f(\sigma, m, \alpha)$, are strongly monotonic (see the discussion in Case 2, as well as property (P1) mentioned above). The theorem follows. □

5 Learning from Fat-Texts and Other Texts

In this section we deal with special kinds of texts. A text is called *fat* iff every datum appears infinitely often in that text. A text T is called *one-one* iff for all $x \in$ content(T), there exists a unique n such that $T(n) = x$. We let fat denote the set of all fat texts and one − one the set of all one-one texts. The main result is given in the following theorem, showing that anything that can be iteratively learnt can be so learnt conservatively, strongly decisively and cautiously (at the same time) *from fat text*. It basically follows from the observation that, on fat text, every learner is sink-locking (see Theorem 8).

First we note that the proof of Theorem 1 also shows that $\mathbf{Txt}^{\mathrm{fat}}\mathbf{ItFex} = \mathbf{Txt}^{\mathrm{fat}}\mathbf{ItEx}$. Furthermore, fat text can always be simulated in the full-information setting, which is the statement of the next lemma. This requires a technical condition which is concerned with "skipping" hypotheses for which we make a definition.

Definition 11. We say that a learning restriction δ *allows for simulation on consistent text* iff, for all $(p, T) \in \delta$, r strictly monotone increasing and T' with $\forall i : \mathrm{content}(T[r(i)]) = \mathrm{content}(T'[i])$, we have $(p \circ r, T') \in \delta$.

Intuitively, $p \circ r$ "skips" some hypotheses which were generated without seeing new information, for example because a learner is simulated by showing many data (which were shown previously). Note that all learning restrictions given in this paper allow for simulation on consistent text, and that the set of all learning restrictions allowing for simulation on consistent text is closed under intersection.

Lemma 12. *Let δ allow for simulation on consistent text. Then we have*

$$\mathbf{Txt}^{\mathrm{fat}}\mathbf{It}\delta\mathbf{Ex} \subseteq \mathbf{TxtG}\delta\mathbf{Ex};$$
$$\mathbf{Txt}^{\mathrm{one-one}}\mathbf{It}\delta\mathbf{Ex} \subseteq \mathbf{TxtG}\delta\mathbf{Ex}.$$

Standard techniques can be used to show the following result.

Theorem 13. $\mathbf{TxtItEx} \subset \mathbf{Txt}^{\mathrm{fat}}\mathbf{ItEx} \subset \mathbf{TxtGEx}$.

Proof. The first inequality is easy to see using the set of languages containing the set of all positive natural numbers and all finite sets of natural numbers *which contain* 0.

Regarding the second inequality, let L and H be a recursively inseparable pair [Rog67]. Consider the class $\mathcal{L} = \{L, \mathbb{N}\} \cup \{L \cup \{x\} : x \in H\}$. $\mathcal{L}$ is $\mathbf{TxtGEx}$-learnable: the learner would first conjecture L and then change to $L \cup \{x\}$ whenever it turns out that some x seen so far is enumerated into H and make another mindchange to $\mathbb{N}$ whenever it turns out that two elements seen in the input are enumerated into H.

However, $\mathcal{L}$ is not $\mathbf{Txt}^{\mathrm{fat}}\mathbf{ItEx}$-learnable. Suppose by way of contradiction that M $\mathbf{Txt}^{\mathrm{fat}}\mathbf{ItEx}$-learns $\mathcal{L}$. Let σ be a locking sequence for M on L (existence of such a σ can be shown for learning from fat texts in a way simlar to the corresponding result for learning from arbitrary texts from [BB75]). Suppose $M^*(\sigma) = e$. Now define a function f as follows: if $M(e, x) = e$ then $f(x) = 1$ else $f(x) = 0$. The function f is total recursive, as $\mathbb{N} \in \mathcal{L}$ and therefore the learner M has to be total, and thus the condition defining f can be evaluated by simulating M. Furthermore, $f(x) = 1$ for all $x \in L$ as σ is a locking sequence for M on L. In addition, $f(x) = 1$ for some $x \in H$, as L and H are not recursively separable. It follows that σ is also a stabilising sequence for M on $L \cup \{x\}$, and thus M does not $\mathbf{Txt}^{\mathrm{fat}}\mathbf{ItEx}$-learn the language $L \cup \{x\}$. Thus, M cannot $\mathbf{Txt}^{\mathrm{fat}}\mathbf{ItEx}$-learn $\mathcal{L}$. □

The above result shows that iterative learners have not only information-theoretic limitations in that they forget past data and cannot recover them (on normal text), but also a computational limitations which cannot be compensated by having fat text. Next we show that fat text always allows for learning conservatively (as well as cautiously and strongly decisively).

Theorem 14. $\mathbf{Txt}^{\text{fat}}\mathbf{ItEx} = \mathbf{Txt}^{\text{fat}}\mathbf{ItConsvEx} = \mathbf{Txt}^{\text{fat}}\mathbf{ItSDecEx}$.

Proof. This follows from Theorem 8 by observing that a fat text always leads to sink-locking of iterative learners. □

The following proposition follows from Lemma 12 and Theorems 2 and 3.

Proposition 15. *(a) There exists a class of languages which is* **TxtItMonEx***,* **TxtItSDecEx***,* **TxtItConsvEx***-learnable but not* $\mathbf{Txt}^{\text{fat}}\mathbf{SMonEx}$*-learnable.*

(b) There is a class which is **TxtItSDecEx***-learnable (and therefore also* **TxtItConsvEx***-learnable) but not* $\mathbf{Txt}^{\text{fat}}\mathbf{ItMonEx}$ *or* $\mathbf{Txt}^{\text{one-one}}\mathbf{ItMonEx}$*-learnable.*

Proof. (a) Let O be the set of odd numbers, and $\mathcal{L} = \{O\} \cup \{\{e \oplus W_e\} : e \in \mathbb{N}\}$. Clearly, $\mathcal{L}$ is **TxtItSDecEx**, **TxtItMonEx**, **TxtItConsvEx**-learnable. However, it is not $\mathbf{Txt}^{\text{fat}}\mathbf{SMonEx}$-learnable, as any learner for it must output on some initial segment σ with $\text{content}(\sigma) \subseteq O$, a grammar for O. Now, σ can be extended to a fat text T for $e \oplus W_e$, where e is such that $W_e = \{x : 2x + 1 \in \text{content}(\sigma)\}$. But then, the learner cannot strongly monotonically learn $e \oplus W_e$ from the text T.

(b) Let $L_e = \{x : x = e \text{ or } e < x \leq |W_e|\}$. Let $\mathcal{L} = \{L_e : e \in \mathbb{N}\}$. It is easy to verify that $\mathcal{L}$ is **TxtItSDecEx**-learnable.

Suppose by way of contradiction that M $\mathbf{Txt}^{\text{fat}}\mathbf{MonEx}$-learns $\mathcal{L}$. Then, by using n-ary recursion theorem, there exist e_1, e_2, e_3 such that $e_1 < e_2 < e_3$ and $W_{e_1} = W_{e_3} = \mathbb{N}$ and W_{e_2} is as defined below. To define W_{e_2}, search for a σ such that $\emptyset \subset \text{content}(\sigma) \subseteq W_{e_3}$ and $W_{M(\sigma)}$ enumerates an element $z > \max(\text{content}(\sigma))$. Once such a σ is found, let W_{e_2} be a set of cardinality $max(\text{content}(\sigma))$. Note that there must exist such a σ as M $\mathbf{Txt}^{\text{fat}}\mathbf{MonEx}$-learns L_{e_3}. Let τ be a extension of σ such that $\text{content}(\tau) = L_{e_2}$, and $M(\tau)$ is a grammar for L_{e_2}. Note that such τ must exist as M $\mathbf{Txt}^{\text{fat}}\mathbf{MonEx}$-learns L_{e_2}. Let T be a fat text for W_{e_1} extending τ. Then, M cannot $\mathbf{Txt}^{\text{fat}}\mathbf{MonEx}$-learn L_{e_1} from T as $z \in W_{M(\sigma)} \cap L_{e_1}$ and $z \notin W_{M(\tau)}$. □

The proof of Theorem 5 can be easily modified to show the following result.

Theorem 16. $\mathbf{TxtItSMonEx} \not\subseteq \mathbf{Txt}^{\text{fat}}\mathbf{ItSNUShEx}$.

We next show that learning from one one-one texts is equivalent to learning from arbitrary text for a number of possible learning restrictions. For giving our result we need the following definition (which is now given in the form for language learning).

Definition 17 (Kötzing [Köt14]). For all $p \in \mathcal{R}$, we let

$$\text{Sem}(p) = \{p' \in \mathcal{R} \mid \forall i : W_{p(i)} = W_{p'(i)}\};$$
$$\text{Mc}(p) = \{p' \in \mathcal{R} \mid \forall i : (p(i) = p(i+1) \Rightarrow p'(i) = p'(i+1))\}.$$

A sequence acceptance criterion δ is said to be a *semantic restriction* iff, for all $(p, g) \in \delta$ and $p' \in \text{Sem}(p)$, $(p', g) \in \delta$. A sequence acceptance criterion δ is said to be a *pseudo-semantic restriction* iff, for all $(p, g) \in \delta$ and $p' \in \text{Sem}(p) \cap \text{Mc}(p)$, $(p', g) \in \delta$.

Intuitively, semantic restrictions allow for arbitrarily changing the syntax of the conjectures, as long as the semantics stay the same. Pseudo-semantic restrictions further require that no additional mind changes are introduced this way.

Note that all learning restrictions given in this paper except **Fex** are pseudo-semantic restrictions.

Theorem 18. *Let δ be a pseudo-semantic restriction allowing for simulation on consistent text (see Definition 11). Then we have, for each set of languages $\mathcal{L}$, $\mathcal{L}$ is* $\mathbf{Txt}^{\text{one-one}}\mathbf{It}\delta$*-learnable iff it is* $\mathbf{TxtIt}\delta$*-learnable.*

Proof. Clearly, if $\mathcal{L}$ is $\mathbf{TxtIt}\delta$-learnable, then it is $\mathbf{Txt}^{\text{one-one}}\mathbf{It}\delta$-learnable.

Now suppose M is a $\mathbf{Txt}^{\text{one-one}}\mathbf{It}\delta\mathbf{Ex}$-learner for $\mathcal{L}$. Define learner N as follows. Intuitively, N will keep track of "elements which caused mind change" by using padding. The initial conjecture of N is $\mathrm{pad}(M(\emptyset), \emptyset)$ and, for all e, D and x,

$$N(\mathrm{pad}(e, D), x) = \begin{cases} \mathrm{pad}(M(e, \#), D), & \text{if } x \in D \cup \{\#\} \text{ or } M(e, x) = e; \\ \mathrm{pad}(M(e, x), D \cup \{x\}), & \text{otherwise.} \end{cases}$$

We now claim that if M $\mathbf{Txt}^{\text{one-one}}\mathbf{ItEx}$-learns L, then N $\mathbf{TxtItEx}$-learns L. To see this, consider any arbitrary text T for L, and consider the behaviour of N on T. Note that for any $x \neq \#$, the second case in the definition of N can apply at most once. Let now T' be the text derived from T such that

- if the second case in the definition of N applies once for x, then replace all except the corresponding occurrence of x in T by $\#$;
- if the second case never applies for x, then replace the first occurrence of x in T by the two symbols x and $\#$ and all other occurrences of x by $\#$.

Now the new text T' as formed above is a one-one text for L, and N *simulates* M on T', possibly skipping ahead with hypotheses whenever an occurrence of x was replaced by $x\#$. The hypotheses output by N are semantically equivalent to those given by M, and new mind changes are not introduced. Thus, N $\mathbf{TxtIt}\delta$-learns $\mathcal{L}$. □

Theorem 19. *There exists a class $\mathcal{L}$ which is* $\mathbf{Txt}^{\text{one-one}}\mathbf{ItFex}$*-learnable but not* $\mathbf{Txt}^{\text{one-one}}\mathbf{Ex}$*-learnable. Therefore $\mathcal{L}$ is not* $\mathbf{TxtItEx}$*-learnable (and hence not* $\mathbf{TxtItFex}$*-learnable).*

Proof. Let $\mathcal{L}$ consist of the languages $L_{e,z}$, $z \leq e$, $e, z \in \mathbb{N}$, where $L_{e,z} = \{(e, x, y) : x = z \text{ or } x + y < |W_e|\}$.

The learner on seeing any input element (e, x, y), outputs a grammar (obtained effectively from (e, x)) for $L_{e,\min(\{e,x\})}$.

If W_e is infinite, then $L_{e,e} = L_{e,z}$ for all $z \leq e$, and thus all the (finitely many) grammars output by the learner are for $L_{e,e}$.

If W_e is finite, then $L_{e,z}$ contains only finitely many elements which are not of the form $(e, z, \cdot)$, and thus on any one-one text for $L_{e,z}$, the learner converges to a grammar for $L_{e,z}$.

We now show that $\mathcal{L}$ is not **TxtEx**-learnable. Suppose otherwise that some learner **TxtEx**-learns $\mathcal{L}$. Then, for $e \geq 2$, W_e is infinite iff the learner has a stabilising sequence [BB75,Ful90] τ on the set $\{(e,x,y) : x,y \in \mathbb{N}\}$ and the largest sum $x+y$ for some (e,x,y) occurring in τ is below $|W_e|$. Thus it would be a Σ_2 condition to check whether W_e is infinite in contradiction to the fact that checking whether W_e is infinite is Π_2 complete. Thus such a learner does not exist. □

Theorem 20. *There exists a class of languages which is iteratively learnable using texts where every element which is maximal so far is marked, but is not* **TxtItEx***-learnable.*

Proof. Let a class $\mathcal{L}$ contain, for all $n \in \mathbb{N}$, the following sets:

$$L_0 = \{2m : m \in \mathbb{N}\},$$
$$L_{2n+1} = \{2m : m \in \mathbb{N}, m \leq n\} \cup \{2n+1\} \text{ and}$$
$$L_{2n+2} = \{2m : m \in \mathbb{N}, m \leq n+1\} \cup \{2n+1\}.$$

To see that $\mathcal{L}$ is iteratively learnable from texts where every maximal element is marked, note that the learner can initially output grammar for L_0. If and when it sees an odd element $2n+1$, it outputs a grammar for L_{2n+1}, if $2n+1$ was the maximal element; otherwise it outputs a grammar for L_{2n+2}. From then on, it changes its mind to L_{2n+2} iff it sees $2n+2$ in the input.

Now we show that $\mathcal{L}$ is not **TxtItEx**-learnable. Suppose by way of contradiction that M **TxtItEx**-learns $\mathcal{L}$. Suppose σ is a locking sequence for M on L_0. Without loss of generality assume that $\text{content}(\sigma) = \{2m : m \leq n\}$ for some n. Now, $M^*(\sigma \diamond 2n+2 \diamond 2n+1 \diamond \#^r) = M^*(\sigma \diamond 2m+1 \diamond \#^r)$, for all r, and thus M fails to identify at least one of L_{2m+1} and L_{2m+2}. □

6 Class Preserving hypotheses spaces

A one-one hypothesis space might be considered in order to prevent that an iterative learner cheats by storing information in the hypothesis. A hypothesis space $(H_e)_{e\in\mathbb{N}}$ is called class preserving (for learning $\mathcal{L}$) iff $\{H_e : e \in \mathbb{N}\} = \mathcal{L}$. A learner is class preserving, if the hypothesis space used by it is class preserving. The following lemma is useful when considering one-one hypothesis spaces.

Lemma 21. *Suppose M* **TxtItEx***-learns $\mathcal{L}$ using one-one class-preserving hypothesis space $\mathcal{H} = \{H_e \mid e \in \mathbb{N}\}$ for $\mathcal{L}$. Then, for all e, for all $x \in H_e$, $M(e,x) = e$.*

Proof. Let σ be locking sequence for M on H_e. Then, since e is the only grammar for H_e, $M^*(\sigma) = e$. Furthermore, $M(e,x) = e$ for all $x \in H_e$. □

The first result shows that the usage of one-one texts increases the learning power of those iterative learners which are forced to use one-one hypothesis spaces, that is, which cannot store information in the hypothesis during the learning process.

Theorem 22. *There exists a class $\mathcal{L}$ having a one-one class preserving hypothesis space such that the following conditions hold:*

(a) $\mathcal{L}$ can be $\mathbf{Txt}^{\text{one-one}}\mathbf{ItEx}$-learnt using any fixed one-one class preserving hypothesis space for $\mathcal{L}$;

(b) $\mathcal{L}$ cannot be $\mathbf{TxtItEx}$-learnt using any fixed one-one class preserving hypothesis space for $\mathcal{L}$.

Proof Sketch. For each e, define L_{2e} and L_{2e+1} based on a recursive enumeration of all pairs of learners and hypothesis spaces, where the e-th pair is $\langle M_e, \mathcal{H}^e \rangle$:

1. Initially, let L_{2e} contain $\{\langle e, 2x\rangle : x \in \mathbb{N}\}$ and L_{2e+1} contain $\{\langle e, 2x+1\rangle : x \in \mathbb{N}\}$.
2. Search for $\sigma_{2e}, \sigma_{2e+1}$ such that
   - content$(\sigma_{2e}) \subseteq \{\langle e, 2x\rangle : x \in \mathbb{N}\}$ and
   - content$(\sigma_{2e+1}) \subseteq \{\langle e, 2x+1\rangle : x \in \mathbb{N}\}$ and
   - $M_e^*(\sigma_{2e}) \neq M_e^*(\sigma_{2e+1})$ and
   - both $\mathcal{H}^e_{M_e^*(\sigma_{2e})}$ and $\mathcal{H}^e_{M_e^*(\sigma_{2e+1})}$ enumerate some (possibly different) element of the form $\langle e, \cdot\rangle$.
3. If and when such σ_{2e} and σ_{2e+1} are found, enumerate the set content$(\sigma_{2e}) \cup$ content(σ_{2e+1}) into both L_{2e} and L_{2e+1}. Search for an element $\langle e, x_e\rangle$ such that $M_e^*(\sigma_{2e} \diamond x_e) \neq M_e^*(\sigma_{2e} \diamond x_e)$.
4. If and when such an x is found, enumerate x in both L_{2e} and L_{2e+1}.

This completes the definition of L_{2e} and L_{2e+1}. Note that by construction $\mathcal{L}$ contains exactly two languages L_{2e} and L_{2e+1} respectively which contain any element of the form $\langle e, \cdot\rangle$.

Now, if the e-th pair $\langle M_e, \mathcal{H}^e\rangle$ witnesses that $\mathcal{L}$ is $\mathbf{TxtItEx}$-learnt by M_e using one-one class preserving hypothesis space $\mathcal{H}^e$ for $\mathcal{L}$, then we get a contradiction as follows. Note that step 1 in the construction of L_{2e} and L_{2e+1} must succeed, as otherwise, M_e does not identify at least one of $L_{2e} = \{\langle e, 2x\rangle : x \in \mathbb{N}\}$ or $L_{2e+1} = \{\langle e, 2x+1\rangle : x \in \mathbb{N}\}$. As $\mathcal{H}^e$ is one-one class preserving hypothesis space for $\mathcal{L}$, we must have that exactly one of $M_e^*(\sigma_{2e})$ and $M_e^*(\sigma_{2e+1})$ are grammars (in hypothesis space $\mathcal{H}^e$) for L_{2e} and the other one is a grammar for L_{2e+1}. Now, step 2 must also succeed to find x_2 as both L_{2e} and L_{2e+1} contain content(σ_{2e}) and content(σ_{2e+1}). But then, both L_{2e} and L_{2e+1} contain x_e and thus M_e violates Lemma 21.

Now we show that $\mathcal{L}$ is $\mathbf{Txt}^{\text{one-one}}\mathbf{ItEx}$-learnable using any fixed one-one class preserving hypothesis space $\mathcal{H} = (H_i)_{i\in\mathbb{N}}$ for $\mathcal{L}$. To see this consider any one-one class preserving hypothesis space for $\mathcal{L}$. Let h_e be the unique grammar for L_{2e} in hypothesis space $\mathcal{H} = (H_i)_{i\in\mathbb{N}}$. Suppose the input element is $\langle e, x\rangle$. Then the learner M searches for a grammar h such that H_h contains $\langle e, x\rangle$, and then outputs h.

It is easy to verify that on any one-one input text T for L_{2e} (respectively L_{2e+1}), the learner M will output $2e$ infinitely often and $2e+1$ only finitely often (respectively, $2e+1$ infinitely often and $2e$ only finitely often). Thus, M $\mathbf{Txt}^{\text{one-one}}\mathbf{ItEx}$-learns $\mathcal{L}$ using hypothesis space $\mathcal{H}$. $\square$

In general, the hierarchy $\mathbf{SMonEx} \subseteq \mathbf{MonEx} \subseteq \mathbf{WMonEx}$ holds. The following result shows that this hierarchy is proper and that one can get the separations even in the case that the more general criterion is made stricter by enforcing the use of a one-one hypothesis space.

Theorem 23. *(a)* **TxtItWMonEx** $\not\subseteq$ **TxtItMonEx***;*
(b) **TxtItMonEx** $\not\subseteq$ **TxtItSMonEx***.*
Here the positive sides can be shown using a one-one class preserving hypothesis space.

Theorem 2 and Theorem 3 show the above result and also provide conservatively learnable families for these separations. We now consider learning by *reliable* learners. A learner is *reliable* if it is total and for any text T, if the learner converges on T to a hypothesis e, then e is a correct grammar for content(T). We denote the reliability constraint on the learner by using **Rel** in the criterion name. For the following result, we assume (by definition) that if a learner converges to ? on a text, then it is not reliable. The next result shows that there is exactly one class which has a reliable iterative learner using a one-one class preserving hypothesis space and this is the class FIN $= \{L : L \text{ is finite}\}$.

Theorem 24. *If $\mathcal{L}$ is* **TxtItRelEx***-learnable using a one-one class preserving hypothesis space then $\mathcal{L}$ must be* FIN*.*

Proof. It is easy to see that FIN is **TxtItRelEx**-learnable using a class preserving one-one hypothesis space.

Now, suppose M is **TxtItRelEx**-learner for $\mathcal{L}$ using one-one class preserving hypothesis space $\mathcal{H} = (H_e)_{e\in\mathbb{N}}$.

If $\mathcal{L}$ contains an infinite language L, then let σ be locking sequence for M on L. Then, M converges on $\sigma \diamond \#^\infty$ to a grammar for L, and thus is not reliable. Thus, $\mathcal{L} \subseteq$ FIN.

Now suppose $\mathcal{L}$ does not contain some finite set S. Let σ be such that content(σ) $= S$. Then, as M does not converge on $\sigma \diamond \#^\infty$, for some r, $M^*(\sigma \diamond \#^r) = e \neq ?$. Now, $M^*(\sigma \diamond \#^r \diamond \#) = e$ (by Lemma 21). Thus M converges on $\sigma \diamond \#^\infty$, a contradiction. □

Theorem 25. *There exists a subclass of* FIN *which is not* **TxtItEx***-learnable using a one-one class preserving hypothesis space.*

Proof. Let $\mathcal{L} = \{L : 2 \leq \text{card}(L) \leq 3\}$. Suppose by way of contradiction that M **TxtItEx**-learns $\mathcal{L}$ using a one-one class preserving hypothesis space $\mathcal{H} = (H_i)_{i\in\mathbb{N}}$.

Note that it is not possible that $M^*(\sigma) = M^*(\tau)$, if content(σ) $\neq$ content(τ), where both content(σ) and content(τ) contain at most 3 elements (otherwise it is easy to verify that M fails to learn $\mathcal{L}$). In particular we may assume without loss generality that M does not output ? on any non-empty input, as we may ignore from consideration elements of the only set S of cardinality at most 3, such that some sequence σ with content(σ) $= S$ may lead to ? output.

Let $a \in \mathbb{N}$ be any element.

Case 1: $M^*(a) = p$, where H_p does not contain a or card(H_p) $= 3$. Then, using Lemma 21, M fails to learn $\{a, b\}$, where $b \in H_p - \{a\}$ from the text $a \diamond b \diamond \#^\infty$.

Case 2: $M^*(a) = p$, where $H_p = \{a, b\}$.

Let $c \notin H_p$.

Case 2.1: $M^*(a \diamond c) = p'$, where $H_{p'} \neq \{a, c\}$. Then, using Lemma 21 M fails to learn $\{a, c\}$ from the text $a \diamond c \diamond \#^\infty$.

Case 2.2: $M^*(a \diamond c) = p'$, where $H_{p'} = \{a, c\}$. Then, using Lemma 21 M fails to learn $\{a, b, c\}$ from the text $a \diamond b \diamond c \diamond \#^\infty$.

Thus, M cannot **TxtItEx**-learn $\mathcal{L}$ using a one-one class preserving hypothesis space. $\square$

Note that in learning theory without loss of generality one assumes that classes are not empty. The next theorem characterises when a class can be iteratively and reliably learnt using a class preserving hypothesis space: it is the case if and only if the set of canonical indices of the languages in the class is recursively enumerable. Note that the hypothesis space considered here is not one-one and that padding is a natural ingredient of the learning algorithm.

Theorem 26. *A class $\mathcal{L}$ has a class-preserving iterative and reliable learner iff it does not contain infinite languages and the set $\{e : D_e \in \mathcal{L}\}$ of its canonical indices is recursively enumerable.*

Proof. It is well known that classes containing infinite languages do not have a reliable learner. Furthermore, it is easy to see that a set D_e is learnt by an iterative reliable learner M using class preserving hypothesis space iff there is a sequence σ with range D_e such that $M(M^*(\sigma), x) = M^*(\sigma)$ for all $x \in D_e \cup \{\#\}$; thus the set of all canonical indices in the class learnt by M is recursively enumerable.

For the converse direction, assume that $\mathcal{L} \subseteq$ FIN is nonempty and recursively enumerable. Now it is shown that $\mathcal{L}$ is **TxtItRelEx**-learnable using the hypothesis space $\mathcal{H}$ consisting of sets $H_{\langle e,s \rangle}$ which are defined as follows: Let L_s denote the elements enumerated into L within s steps; if $e \in L_s$ then $H_{\langle e,s \rangle} = D_e$ else $H_{\langle e,s \rangle}$ is some default finite set in $\mathcal{L}$. For this, note that the set D_e is defined such that $e = \sum_{x \in D_e} 2^x$ and therefore $D_0 = \emptyset$.

The iterative learner always conjectures indices of the form $\langle e, s \rangle$ where e is maintained such that D_e contains all the data observed so far and s is a parameter which is used to enforce syntactic divergence in the case that e is not yet enumerated into L and which either grows or stabilises. The initial hypothesis is $\langle 0, 0 \rangle$. Given an old hypothesis $\langle d, s \rangle$ and observing datum x, the new hypothesis is computed as follows:

- Let e be the unique index with $D_e = D_d \cup \{x\} - \{\#\}$;
- If $e \in L_s$ then the new hypothesis is $\langle e, s \rangle$ else the new hypothesis is $\langle e, s + 1 \rangle$.

It is easy to see by induction that the current hypothesis always is a pair $\langle e, s \rangle$ such that D_e consists of all the data observed so far. In the case that the set to be learnt is infinite, the parameter e will grow unboundedly and the sequence of hypotheses is therefore divergent and reliability is assured. In the case that the set to be observed is finite, then from some time on the parameter e will stabilise at the correct value. In the case that $e \notin L$ the parameter s will – after the correct value for e is reached – increase in every step and the learner will diverge. In the case that $e \in L$ the parameter s will stop growing once $e \in L_s$ is reached and from that point onwards the learner has converged to a hypothesis $\langle e, s \rangle$ such that D_e is the set to be learnt and $e \in L_s$; therefore $H_{\langle e,s \rangle} = D_e$ and the hypothesis is correct.

In summary, the learner converges to some hypothesis $\langle e, s \rangle$ if and only if the set to be learnt is finite and in the class to be learnt; furthermore, in the case of convergence, $H_{\langle e,s \rangle} = D_e$ and D_e

is equal to the set to be learnt. As the hypothesis space $\mathcal{H}$ is class preserving, the given learner satisfies all required conditions. □

7 Syntactic versus Semantic Conservativeness

A learner is called *semantically conservative* iff whenever it outputs two indices i, j such that $W_i \neq W_j$ and i is output before j then the hypothesis j is based on some observed data not contained in W_i. This notion coincides with syntactic conservative learning in the case of standard explanatory learning; however, in the special case of iterative learning, it is more powerful than the usual notion of conservative learning.

Theorem 27. *There is a class $\mathcal{L}$ which can be learnt iteratively and strongly monotonically and semantically conservatively but which does not have an iterative and syntactically conservative learner.*

Proof. Let the class $\mathcal{L}$ consist of the following languages constructed for each n:

- First one constructs a text T_n starting with $3n+1 \diamond 0$ and then extended by sequences of numbers of the form $3m$ such that the text is extended by a new piece whenever this new piece causes the n-th iterative learner M_n to make a mind change. L_{3n} is the content of this text T_n (or the finite part of it constructed).
- In the case that T_n is a complete text then L_{3n+1} and L_{3n+2} are equal to L_{3n}.
- In the case that only a finite part σ_n of T_n is constructed, let m be the canonical index for this sequence σ_n. Let L_{3n+1} consists of content$(\sigma_n) \cup \{3m+2\}$ plus the first number $3h$ found (if any) such that there is a finite sequence $\tau_n \in 0^*$ for which M_n outputs on $\sigma \diamond 3h \diamond 3m+2 \diamond \tau_n$ and $\sigma \diamond 3h \diamond 3m+2 \diamond \tau_n \diamond 0$ the same hypothesis e_n while it outputs a different index on $\sigma \diamond 3h \diamond 3m+2 \diamond \tau_n \diamond 3h$. Furthermore, L_{3n+2} consists of $3n+1, 3m+2$ and all numbers of the form $3k$.

Now one shows that no learner M_n iteratively and syntactically conservatively learns the class $\mathcal{L}$. First, in the case that the text T_n is total, the learner M_n fails to converge on this text for L_{3n} and therefore does not learn the set.

Second, in the case that only a finite part σ_n of T_n is constructed and $3h$ is not enumerated into L_{3n+1} then either M_n does not converge on the text $\sigma \diamond 3m+2 \diamond 0^\infty$ for L_{3n+1} or it converges to a hypothesis e_n which is later not revised when seeing any number of the form $3k$. In the first subcase (the nonconvergence) the learner fails to learn the set L_{3n+1} and in the second subcase the learner converges on some texts for L_{3n+1} and L_{3n+2} to the same index and fails to learn one of these sets.

Third, in the case that only a finite part σ_n of T_n is constructed and $3h$ is enumerated into L_{3n+1}, then the witnesses for this enumeration testify that either e_n is not the correct index (although the learner converges on the text $\sigma_n \diamond 3h \diamond 3m+2 \diamond \tau_n \diamond 0^\infty$ for L_{3n+1} to this index) or there is a mind change witnessing that M_n is not syntactically conservative on the text

$\sigma_n \diamond 3h \diamond 3m+2 \diamond \tau_n \diamond 3h \diamond 0^\infty$ for L_{3n+1}. This case-distinction completes the proof that each M_n fails to learn the class in an iterative and syntactically conservative manner.

Furthermore, an iterative and strongly monotonic learner can be made by conjecturing $\emptyset$ until either the number $3n+1$ or $3m+2$ are revealed (note that the latter one codes the number $3n+1$ by coding σ_n). In the first case the learner conjectures the set L_{3n} whose index can be computed from n. In the second case and or whenever later $3m+2$ has appeared in the input, the learner updates its conjecture to L_{3n+1} as an r.e. index for this language can be computed from m. If the learner sees one number $3k$ outside $\text{content}(\sigma_n) \cup \{3m+2\}$ then the learner pads this number into the previous hypothesis without making a semantic mind change. If it sees a further number $3k'$ outside $\text{content}(\sigma_n) \cup \{3k, 3m+2\}$ then the learner updates to a hypothesis for L_{3n+2} which again can be computed from m. Note that these updates are all strongly monotonic as long as the learner sees only data from sets in the class and that this is sufficient in the present work, where all learning criteria considered are only required for texts of languages inside the given class to be learnt. On data not belonging to any languages of the class, the strongly monotonic behaviour is not guaranteed. Furthermore, note that given the choices of the algorithm, all updates are semantically conservative. The element $3m+2$ guarantees that L_{3n+1} is a proper superset of L_{3n} and that furthermore it has at most one element outside $\text{content}(\sigma_n) \cup \{3m+2\}$ and therefore also the second semantic mind change is semantically conservative. $\square$

References

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

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

[BCMSW08] Ganesh Baliga, John Case, Wolfgang Merkle, Frank Stephan and Rolf Wiehagen. When unlearning helps. *Information and Computation*, 206:694–709, 2008.

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

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

[Cas99] John Case. The power of vacillation in language learning. *SIAM Journal on Computing*, 28:1941–1969, 1999.

[CK10] John Case and Timo Kötzing. Strongly non-U-shaped learning results by general techniques. *Proceedings of COLT (Conference on Computational Learning Theory)*, pages 181–193, 2010.

[CL82] John Case and Chris Lynes. Machine inductive inference and language identification. *Proceedings of ICALP (International Colloquium on Automata, Languages and Programming)*, pages 107–115, 1982.

[CM08a] John Case and Samuel E. Moelius. Optimal language learning. *Proceedings of ALT (Algorithmic Learning Theory)*, pages 419–433, 2008.

[CM08b] John Case and Samuel E. Moelius. U-shaped, iterative, and iterative-with-counter learning. *Machine Learning*, 72:63–88, 2008.

[Ful90] Mark Fulk. Prudence and other conditions on formal language learning. *Information and Computation*, 85:1–11, 1990.

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

[GL04] Gunter Grieser and Steffen Lange. Incremental learning of approximations from positive data. *Information Processing Letters*, 89:37-42, 2004.

[Jan91] Klaus-Peter Jantke. Monotonic and non-monotonic inductive inference of functions and patterns. J. Dix, K. Jantke, and P. Schmitt, editors, *Nonmonotonic and Inductive Logic*, volume 543 of *Lecture Notes in Computer Science*, pages 161–177. 1991.

[JMZ13] Sanjay Jain, Samuel E. Moelius and Sandra Zilles. Learning without coding. *Theoretical Computer Science*, 473:124–148, 2013.

[JORS99] Sanjay Jain, Daniel Osherson, James Royer and Arun Sharma. *Systems that Learn: An Introduction to Learning Theory.* MIT Press, Cambridge, Massachusetts, second edition, 1999.

[Köt09] Timo Kötzing. *Abstraction and Complexity in Computational Learning in the Limit.* PhD thesis, University of Delaware, 2009. Available online at http://pqdtopen.proquest.com/#viewpdf?dispub=3373055.

[Köt14] Timo Kötzing. A Solution to Wiehagen's Thesis. *Symposium on Theoretical Aspects of Computer Science*, (STACS 2014), pages 494–505, 2014.

[LG02] Steffen Lange and Gunter Grieser. On the power of incremental learning. *Theoretical Computer Science*, 288:277–307, 2002.

[LG03] Steffen Lange and Gunter Grieser. Variants of iterative learning. *Theoretical Computer Science*, 292:359–376, 2003.

[LZ93] Steffen Lange and Thomas Zeugmann. Monotonic versus non-monotonic language learning. *Proceedings of Nonmonotonic and Inductive Logic*, pages 254–269, 1993.

[LZ96] Steffen Lange and Thomas Zeugmann. Incremental learning from positive data. *Journal of Computer and System Sciences*, 53:88–103, 1996.

[LZZ08] Steffen Lange, Thomas Zeugmann and Sandra Zilles. Learning indexed families of recursive languages from positive data: a survey. *Theoretical Computer Science*, 397:194–232, 2008.

[OSW82] Daniel Osherson, Micheal Stob and Scott Weinstein. Learning strategies. *Information and Control*, 53:32–51, 1982.

[OSW86] Daniel Osherson, Micheal Stob and Scott Weinstein. *Systems that Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists.* MIT Press, Cambridge, Mass., 1986.

[OW82] Daniel Osherson and Scott Weinstein. Criteria of language learning. *Information and Control*, 52:123–138, 1982.

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

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

[Wie91] Rolf Wiehagen. A thesis in inductive inference. *Proceedings of the Workshop on Nonmonotonic and Inductive Logic*, pages 184–207, 1991.