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

In this paper we consider *inductive inference*, a branch of algorithmic learning theory. This branch analyses the problem of algorithmically learning a description for a formal language (a recursively enumerable subset of the set of natural numbers) when presented successively all and only the elements of that language. For example, a learner M might be presented more and more even numbers. After each new number, M outputs a description for a language as its conjecture. The learner M might decide to output a program for the set of all multiples of 4, as long as all numbers presented are divisible by 4. Later, when M sees an even number not divisible by 4, it might change this guess to a program for the set of all multiples of 2.

Gold, in his seminal paper [Gol67], introduced this idea of learning a language in the limit formally. His first and simple learning criterion was **TxtGEx**-*learning*[4], where a learner is *successful* iff, on every *text* for L (listing of all and only the elements of L) it eventually stops changing its conjectures, and its final conjecture is a correct description (an *explanation*) for the input sequence. Trivially, each single, describable language L has a suitable constant function as a **TxtGEx**-learner (this learner constantly outputs a description for L). Thus, we are interested

⋆ Supported by NUS grants C252-000-087-001 and R146-000-181-112.
⋆⋆ Supported in part by NUS grant R146-000-181-112.
[4] **Txt** stands for learning from a *text* of positive examples; **G** stands for Gold, who introduced this model, and is used to indicate full-information learning; **Ex** stands for *explanatory*.

in analyzing for which *classes of languages* $\mathcal{L}$ there is a *single learner* h learning *each* member of $\mathcal{L}$. This framework is also sometimes known as *language learning in the limit* and has been studied extensively, using a wide range of learning criteria similar to **TxtGEx**-learning (see, for example, the textbook [JORS99]).

It is easy to see from the definition of **TxtGEx**-learning that the learner can be arbitrarily inefficient: the learner can postpone computations and decisions until more data has been shown; no restrictions on the computing time (beyond linear) in each update step will restrict the learner's abilities. One way to address this problem is to restrict the access to past data. The most common formalization of this idea is *iterative learning* (**TxtItEx**) [Wie76], where the learner, in each iteration, gets to see only the new data item and its previous hypothesis. Due to the padding lemma, this memory of the previous hypothesis is still not void, but finitely many data can be memorised by padding the hypothesis. In effect, syntactic changes of the hypothesis, which do not affect its semantics, are used as a memory.

There are several approaches which aim to make updates more meaningful. One direction is to consider one-one hypothesis spaces where the learner cannot do padding without changing the semantics of the previous hypothesis. Other restrictions on the updates are requiring that they respect some semantic constraints towards preserving already achieved quality of the previous hypothesis and avoiding obvious errors. For example,

- updates have to be motivated by inconsistent data observed (syntactic conservativeness) [Ang80,OSW86],
- semantic updates have to be motivated by inconsistent data observed (semantic conservativeness) [GJS13],
- updates cannot repeat semantically abandoned conjectures (decisiveness) [OSW82],
- updates cannot go from correct to incorrect hypotheses (non-U-shapedness) [BCMSW08],
- conjectures cannot be proper supersets of the language to be learnt (cautiousness) [OSW86] or
- conjectures have to contain all the data observed so far (consistency) [Bar74].

In particular those constraints in the list which rule out updates without a semantic improvement of the hypothesis do in some cases effectively hinder padding and are therefore restrictive compared to plain iterative learning.

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, the work in this area had two shortcomings: (a) it was not clear how strong an update restriction is necessary to actually restrict the learning power below full iterative learning; and (b) there was no complete picture of the relations of the mentioned update restrictions in the setting of iterative learning. With this paper we eliminate these shortcomings: Regarding (a), we show that strong decisiveness, but not decisiveness restricts iterative learning in learning power. Regarding (b) we completely characterise the relationship of the iterative learning criteria with the different restrictions. This is depicted in the diagramme in Figure 1, representing a *map* of the update constraints and their relations. A black line between two learning criteria indicates a trivial inclusion (where the inclusion follows directly from the definition of the restriction). A gray box around criteria indicates equality of these criteria, as found in this work.

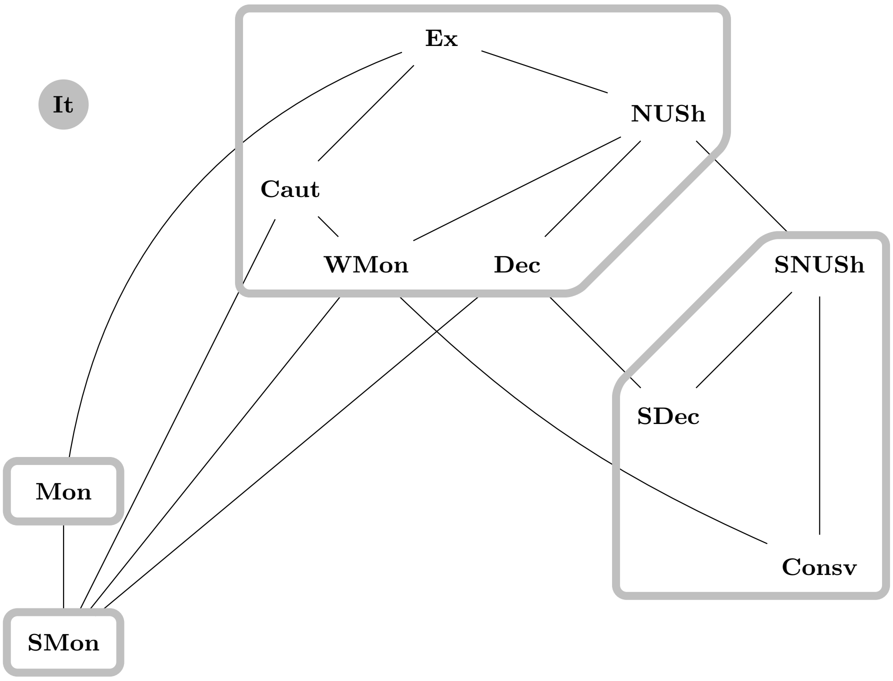


Fig. 1. Relation of criteria combined with iterative learning.

Our work extends a breakthrough result by Case and Moelius [CM08b] who showed that iterative learners can be made non-U-shaped. The present work improves this result by showing that iterative learners 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]. This result is given in Theorem 10 in Section 4. Also in that section are the other results giving the complete characterisation indicated in Figure 1.

Further sections give additional results, complementing the statements shown in Section 4. Section 5 considers alternative text-types, such as *fat texts* and *one-one texts*. Here a text is *fat* if every datum appears infinitely often and is *one-one* if every datum appears exactly once. It is interesting to see that, for iterative learning from fat texts, the divide between decisive and strongly decisive learning vanishes and instead neither update constraint restricts the learning power of iterative learning. Section 6 considers class preserving hypothesis spaces while Section 7 considers a semantic variant of conservative learning.

We proceed with Section 2, which gives mathematical definitions, followed by Section 3 which formally defines learning criteria.

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. For two sets A and B, their join is defined as: $A \oplus B = \{2x \mid x \in A\} \cup \{2x + 1 \mid x \in B\}$. Let D_e denote the finite set with canonical index e: that is, $e = \sum_{x \in D_e} 2^x$. Note that $D_0 = \emptyset$.

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 $\lambda x_1, \ldots, x_n \centerdot f(x_1, \ldots, x_n)$. For example, with $c \in \mathbb{N}$, $\lambda x \centerdot c$ 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 $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$. We can extend the pairing function to triple by using $\langle x, y, z\rangle = \langle x, \langle y, z\rangle\rangle$. One can similarly code any tuples. 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, $W_{\mathrm{pad}(e,D)} = W_e$.

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 Kötzing [Köt09] with his "building-blocks" approach for defining learning criteria.

A *language* is an r.e. set $L \subseteq \mathbb{N}$. Any total function $T : \mathbb{N} \to \mathbb{N} \cup \{\#\}$ is called a *text*. A text can also be considered as an infinite sequence of elements. Intuitively, $\#$ denotes pause in the presentation of data, that is, "no new input data in the text." For each (possibly infinite) sequence q with its range contained in $\mathbb{N} \cup \{\#\}$, let $\mathrm{content}(q) = (\mathrm{range}(q) \setminus \{\#\})$. For any given language L, a *text for* L is a text T such that $\mathrm{content}(T) = L$. We let σ and τ range over initial segments of texts. The length of σ is denoted by $|\sigma|$. We let $T[n]$ denote the initial segment of T of length n, and for $n \geq |\sigma|$, $\sigma[n]$ the initial segment of σ of length n. We let SEQ denote the set of all initial segments of all texts. Concatenation of two initial segments σ and τ is denoted by $\sigma \diamond \tau$. If $\sigma \subseteq \tau$ (respectively, $\sigma \subseteq T$), then we also say that τ extends σ (respectively, T extends σ). For a given set of texts F, we let $\mathbf{Txt}^F(L)$ denote the set of all texts in F for L.

A *learner* is a partial recursive function from SEQ to $\mathbb{N} \cup \{?\}$. Intuitively, one can consider a learner M as being given a text as an input, one element at a time, and $M(T[n])$ as the learner's hypothesis after having received the input $T[n]$. Intuitively, ? denotes "no change in hypothesis", that is, the learner repeats its previous hypothesis (or in the case of no earlier hypothesis, the learner does not have enough information to form a hypothesis). By using an appropriate coding, we assume that ? and # can be handled by partial recursive functions.

In special cases, such as for *iterative* learners [Wie76], we consider a modification of the above definition of learners. Intuitively, a learner as defined above is iterative if its new conjecture depends only on its previous conjecture and the new datum, that is, for all σ, $\tau \in$ SEQ and $x \in \mathbb{N} \cup \{\#\}$: $M(\sigma) = M(\tau)$ implies $M(\sigma \diamond x) = M(\tau \diamond x)$. Thus, for ease of notation and based on convention, we consider iterative learners as receiving two inputs, the previous conjecture and the new datum, and outputting a new conjecture. That is, an iterative learner is a mapping from $(\mathbb{N} \cup \{?\}) \times \mathrm{SEQ}$ to $(\mathbb{N} \cup \{?\})$. The initial conjecture of the iterative learner is denoted by $M(\emptyset)$.

More precisely, one can formalise these concepts using an *interaction operator*, which 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. Gold [Gol67] started the study of general learners, wheras iterative learners were first considered by Wiehagen [Wie76]. 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.

We say that M made a mind change at $T[n+1]$, if $M(T[n+1]) \neq M(T[n])$ (respectively, $M^*(T[n+1]) \neq M^*(T[n])$ for iterative learners).

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,

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

Furthermore, we formally define the restrictions discussed in Section 1 in Figure 2. Consistency (**Cons**) was first considered by Bārzdins [Bar74], Conservativeness (**Consv**) was introduced by Angluin [Ang80], and cautiousness (**Caut**) was first considered by Osherson, Stob and Weinstein [OSW86]. Decisiveness (**Dec**) was introduced by Osherson, Stob and Weinstein [OSW82], whereas Non-U-shapedness (**NUSh**) was first studied by Baliga et. al. [BCMSW08]. Study of monotonicity requirements in conjectures is motivated by the fields of monotonic and non-monotonic logic. Strong monotonicity (**SMon**) and Weak monotonicity (**WMon**) were introduced by [Jan91], and monotonicity (**Mon**) was introduced by [Wie90].

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

Fig. 2. Definitions of learning restrictions.

We combine 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

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

and we use $\mathbf{Txt}^F\beta\delta$ to denote the set of all $\mathbf{Txt}^F\beta\delta$-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. In this paper, it will always be assumed that the hypothesis spaces are uniformly r.e., that is, $\{\langle e, x\rangle \mid x \in H_e\}$ is an r.e. set.

A sequence σ is said to be a *locking sequence* [BB75] for M on a language L iff (a) content$(\sigma) \subseteq L$, (b) for all τ such that content$(\tau) \subseteq L$, $M(\sigma) = M(\sigma \diamond \tau)$, and (c) $M(\sigma)$ is a grammar for L (in the hypothesis space used by the learner M). If σ only satisfies (a) and (b) above, then it is called a *stabilising sequence* [Ful90] for M on L.

When considering iterative learners, we use the definition of locking sequence with respect to M^*, but for ease of notation still call σ to be a locking sequence for M on L.

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. Intuitively, the new learner N is constructed as follows: N keeps track of all the past conjectures of M and does not change its mind if M changes its mind to a conjecture made in the past. The conjectures of N are of the form pad(p, D), where D is a finite set of conjectures made earlier by N.

Now, we formally and more precisely define N. $N(\emptyset) = \text{pad}(i, \emptyset)$, where $i = M(\emptyset)$, is the initial conjecture of M. If $M(p, x) \in D \cup \{p\}$, then let $N(\text{pad}(p, D), x) = \text{pad}(p, D)$, else let $N(\text{pad}(p, D), x) = \text{pad}(M(p, x), D \cup \{p\})$.

Now, it is shown that N **TxtItEx**-learns $\mathcal{L}$. Consider a text $T = x_0 \diamond x_1 \diamond x_2 \diamond \ldots$ for a language $L \in \mathcal{L}$. We will define below another text $T' = x_0 \diamond \tau_0 \diamond x_1 \diamond \tau_1 \ldots$ such that for all n,

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

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

Note that (E1) holds by definition for $n = 0$. So suppose we have defined $\tau_0, \tau_1, \ldots, \tau_{m-1}$ and (E1) holds for all $n \leq m$. Then, consider $n = m + 1$. Suppose $N^*(x_0 \diamond x_1 \diamond \ldots \diamond x_{m-1}) = \text{pad}(p, D)$. 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 $N^*(x_0 \diamond x_1 \diamond \ldots \diamond x_{m'}) = \text{pad}(M(p, x_m), D')$, for some finite set D'; 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 $x_0, x_1, \ldots, x_{m-1}$.

Let $T' = x_0 \diamond \tau_0 \diamond x_1 \diamond \tau_1 \ldots$; note that T' is also a text for content$(T) = L$. For $i \in \mathbb{N}$, let p_i, d_i be such that $\text{pad}(p_i, D_{d_i}) = N^*(x_0 \diamond x_1 \diamond \ldots \diamond x_{n-1})$. Now it follows from the definition of N and (E1) that, for all i,

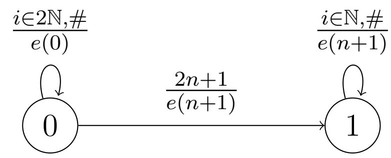
(i) $D_{d_i} \subseteq D_{d_{i+1}} \subseteq D_{d_i} \cup \{p_i\}$,
(ii) $p_0, p_1, \ldots$ is a subsequence of the sequence of conjectures of M on T' and
(iii) if $p_i \neq p_{i+1}$, then $d_{i+1} \neq d_i$.

By (ii) and M **TxtItFex**-learning L, it follows that $\{p_0, p_1, \ldots\}$ is finite, and thus by (i), $d_0, d_1, \ldots$ converges. Thus, by (iii) the sequence $p_0, p_1, \ldots$ also converges. As M **TxtItFex**-learns L, by (ii), all but finitely many elements in the sequence $p_0, p_1, \ldots$ are grammars for L. Thus, N **TxtItEx**-learns L. □

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. The proofs of the next two theorems are based on the proofs of **TxtMonEx** $\not\subseteq$ **TxtSMonEx** and **TxtWMonEx** $\not\subseteq$ **TxtMonEx** from [LZ93].

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 $\mathcal{L} = \{L_n \mid n \in \mathbb{N}\}$. 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).



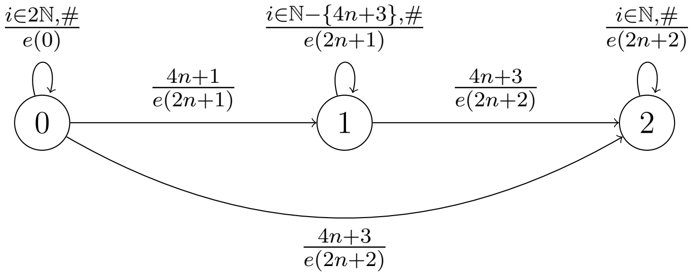
Clearly, M is a **TxtItSDecConsvMonCautEx**-learner for $\mathcal{L}$. The class $\mathcal{L}$ is not strongly monotonically learnable. To see this, suppose by way of contradiction that a learner N **TxtGSMonEx**-learns $\mathcal{L}$. Suppose N conjectures grammar for L_0 after having seen some input σ with content$(\sigma) \subseteq L_0$ (note that there must exist such a σ). Let n be larger than any element in content(σ). Then, for any text T for L_{n+1}, for some τ such that $\sigma \subseteq \tau \subseteq \sigma \diamond T$, $N(\tau)$ must be a grammar for L_{n+1}. But then, N is not strongly monotonic as it changed its conjecture from L_0 to L_{n+1}, which is not a superset for L_0. □

Note that one can modify the protocol in the above proof such that M only memorises the state (and not its conjecture); 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 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 fulfills all the desired requirements for **TxtItSDecConsvCautEx**-learning $\mathcal{L}$. To see that $\mathcal{L}$ is not in **TxtGMonEx**, suppose by way of contradiction that N witnesses that $\mathcal{L}$ is in **TxtGMonEx**. Let σ be such that $\mathrm{content}(\sigma) \subseteq L_0$ and $N(\sigma)$ is a grammar for L_0 (note that there exists such a σ). Let n be larger than any element in $\mathrm{content}(\sigma)$. Then, let τ be such that $\mathrm{content}(\tau) \subseteq L_{2n+1}$, and $N(\sigma \diamond \tau)$ is a grammar for L_{2n+1} (note that there must exist such a τ as N learns L_{2n+1}). Furthermore, let τ' be such that $\mathrm{content}(\tau') \subseteq L_{2n+2}$ and $N(\sigma \diamond \tau \diamond \tau')$ is a grammar for L_{2n+2} (again, note that there must exist such a τ' as N learns L_{2n+2}). But then, N is not monotonic on L_{2n+2}, as $N(\sigma)$ contained $2n+2$, $N(\sigma \diamond \tau)$ does not contain $2n+2$, but $N(\sigma \diamond \tau \diamond \tau')$ again contains $2n+2$ which is a member of L_{2n+2}. $\square$

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. $\square$

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.

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 and $d \in \mathbb{N}$ (where $F(\cdot)$, $G(\cdot)$ are recursively enumerable sets in the parameters described later):

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

Let α_s denote the sequence $1 \diamond \# \diamond 3 \diamond \# \ldots \diamond 2s+1$. Now we define the sets $F(e)$ and $G(e,d)$ based on the following cases.

(a) If there exists an s such that $M_e^*(4e \diamond \alpha_s) = M_e^*(4e \diamond \alpha_{s'})$, for all $s' > s$, then $F(e) = \{0, 1, 2, \ldots, s\}$, for the least such 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 $k > d$ and $r \in \mathbb{N}$ such that $M_e^*(4e \diamond \alpha_{\max(F(e))} \diamond \# \diamond 4d+2 \diamond \#^r) = M_e^*(4e \diamond \alpha_{\max(F(e))} \diamond \# \diamond 4d+2 \diamond \#^r \diamond \#) \neq M_e^*(4e \diamond \alpha_{\max(F(e))} \diamond \# \diamond 4d+2 \diamond \#^r \diamond \# \diamond 2k+1)$, 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 **TxtItSMonEx** 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 $(4e \diamond \alpha_{\max(F(e))} \diamond \# \diamond 2k+1 \diamond \# \diamond 4d+2 \diamond \#^r)$ and $(4e \diamond \alpha_{\max(F(e))} \diamond \# \diamond 2k+1 \diamond \# \diamond 4d+2 \diamond \#^r \diamond 2k+1)$ and first of them must be a correct hypothesis for $\{2e, 2d+1\} \oplus G(e,d)$, as otherwise the learner M_e does not learn it from the text — $4e \diamond \alpha_{\max(F(e))} \diamond \# \diamond 2k+1 \diamond \# \diamond 4d+2 \diamond \#^r \diamond \#^\infty$ — 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* **TxtItConsvEx**, **TxtItSDecEx** *and* **TxtItWMonEx**-*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 a one-one recursive function 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 using the S-m-n Theorem we let $g \in \mathcal{R}$ be a one-one recursive function such that, for all σ,

$$W_{g(\sigma)} = \begin{cases} \emptyset, & \text{if } \sigma = \emptyset \text{ (Case 1);} \\ \{x \mid x \leq 2|\sigma|\} \setminus \{\sigma(0)\}, & \text{if } \sigma \neq \emptyset \text{ and } N^*(\sigma) \neq N^*(\sigma \diamond \#) \text{ (Case 2);} \\ W_{N^*(\sigma)} \cup \mathrm{content}(\sigma), & \text{otherwise (Case 3).} \end{cases}$$

We let O be an iterative learner with initial conjecture $g(\emptyset)$ and, given the previous 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 iterative and 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, either $\sigma = \emptyset$ or $\sigma(0) \neq \#$.

Now consider any two distinct hypothesis $g(\sigma)$ and $g(\sigma \diamond y \diamond \tau)$ output by O. If $\sigma = \emptyset$, then clearly, $y \neq \#$ and $W_{g(\sigma)} \neq W_{g(\sigma \diamond y \diamond \tau)}$ by definition of g. If both $W_{g(\sigma)}$ and $W_{g(\sigma \diamond y \diamond \tau)}$ are defined via Case 2 in the definition of g, then clearly, these are semantically different hypothesis, as $\{z \mid z \leq 2|\sigma|\} \setminus \{\sigma(0)\} \neq \{z \mid z \leq 2|\sigma \diamond y \diamond \tau|\} \setminus \{\sigma(0)\}$. If $\sigma \neq \emptyset$, and exactly one of $W_{g(\sigma)}$

and $W_{g(\sigma \diamond y \diamond \tau)}$ is defined via Case 2 in the definition of g, then again, one of them contains $\sigma(0)$, while the other does not and so these hypotheses are semantically different. Now, consider the remaining case of both $g(\sigma)$ and $g(\sigma \diamond y \diamond \tau)$ being defined via Case 3 in the definition of g. Thus, $N^*(\sigma) = N^*(\sigma \diamond \#)$, $N^*(\sigma \diamond y \diamond \tau) = N^*(\sigma \diamond y \diamond \tau \diamond \#)$, $y \neq \#$, and $g(\sigma)$ and $g(\sigma \diamond y \diamond \tau)$ are both defined via Case 3 in the definition of g. Note that this also implies that O at some point conjectures $g(\sigma \diamond y)$ and thus $N^*(\sigma) \neq N^*(\sigma \diamond y)$. We now have that $y \in W_{g(\sigma \diamond y \diamond \tau)} \setminus W_{g(\sigma)}$ (y 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. $\mathbf{TxtItSDecEx} = \mathbf{TxtItConsvEx} = \mathbf{TxtItSNUShEx}$.

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 $\mathbf{TxtItNUShEx} = \mathbf{TxtItEx}$; we finally show that this proof can be extended to also cover decisiveness, weak monotonicity and caution.

Here is a brief reason for the complications in doing the simulation and the intuition behind our proof. In non-iterative learning, often simulations are done by searching for a locking sequence. However, in iterative learning, it is not possible to search for a locking sequence as iterative learners forget data. A standard trick to address this, for an iterative simulator N to simulate an iterative learner M, on input text T, is to keep track of γ_n defined as follows: $\gamma_0 = \emptyset$ and, for $n > 0$, if $M^*(\gamma_{n-1} \diamond T(n-1)) \neq M^*(\gamma_{n-1})$, then $\gamma_n = \gamma_{n-1} \diamond T(n-1)$, else $\gamma_n = \gamma_{n-1}$. That is, γ's are constructed to keep track of elements which caused a mind change in M. It is easy to verify that $M^*(\gamma_n) = M^*(T[n])$, and the sequence $(\gamma_n)_{n \in \mathbb{N}}$ converges. The above allows one to obtain in the limit a grammar for the target language.

For maintaining properties such as weak monotonicity or decisiveness, one would like to constrain or spoil intermediate conjectures output before the final hypothesis. Thus, one would like to simulate $W_{M^*(\gamma_n)}$, and if M changes its mind on some extension $\gamma_n \diamond \tau$, with $\mathrm{content}(\tau) \subseteq W_{M^*(\gamma_n)}$, we would like to "stop further enumeration" in the conjecture of N obtained from γ_n. In iterative learning this causes the following problem: It is possible that some element x causes a mind change for M if given right after γ_n, but not if given after some proper initial segment of γ_n. Thus, the learner N may not see x in the future in the input text T after having seen $T[n]$ (and may have forgotten having seen it earlier). Thus, it cannot safely spoil the conjecture corresponding to γ_n. To address this, one would like to ignore such x, pretending that they had come earlier and forgotten. However, this works only if there are only finitely many such x, and causes problems for learning infinite languages when there maybe infinitely many such x.

To address this, in the proof used below (which gets its inspiration from the work of Case and Moelius [CM08b] for non-U-shaped iterative learning), we will use a bound m_n on the x's which we tentatively ignore as above. Thus, we will split γ_n as above into σ_n and α_n, where elements in content(α_n) which cause a mind change for M when given right after σ_n will be bounded by m_n (for technical ease, for maintaining some length properties, we may actually have some extra elements in $\sigma_n \diamond \alpha_n$, which do not cause any mind change for M when inserted at that position). The value of m_n may be increased when it is safe to do so (see case (iv) in the definition of N in the proof below). The details of when it is safe are complicated and are stated more formally in the proof below. On the other hand, when N gets as input an element x which is not safe to ignore (as it causes a mind change on all initial segments of σ_n, or is too big etc.), we will reset σ_{n+1} to be $\sigma_n \diamond \alpha_n \diamond x$, α_{n+1} to $\emptyset$ and m_{n+1} to 0 (see cases (ii) and (v) in the definition of N). In case (iii) of the definition of N it is tentatively safe to ignore x as it is bounded by m_n or does not cause a mind change. In case (vi) of the definition of N it is safe to ignore the input x as it does not cause any mind change nor gives enough time to discover a harmful mind change using the previous conjecture. The conjecture corresponding to the parameters σ_n, m_n, α_n used by N is $f(\sigma_n, m_n, \alpha_n)$.

The above almost works, but we need additionally some tricks to make sure that learnability happens. In some cases we do not spoil/constrain the conjecture $f(\sigma_n, m_n, \alpha_n)$ completely on seeing a potentially harmful $x \in W_{M^*(\gamma_n)}$, but just temporarily suspend the simulation until we discover that future conjectures of N on the language $W_{M^*(\gamma_n)}$ also enumerate elements of $W_{M^*(\gamma_n)}$: then it is safe to enumerate these elements (see step 2(b) in the definition of f in the proof). We now proceed with the formal result.

Theorem 10. **TxtItEx** $=$ **TxtItDecEx** $=$ **TxtItWMonEx** $=$ **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 (decisive, cautious) and learns $\mathcal{L}$. Let

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

Intuitively, $B_M(\sigma)$ and $C_M(\sigma)$ respectively denote the set of elements on which M changes/does not change its mind, when these elements are received by M just after σ. $B_M^{\cap}(\sigma)$ is then the set of elements on which M would make a mind change, when received after any initial segment of σ. Members of $B_M^{\cap}$ are thus crucial in the simulation and cannot be ignored. $CB_M(\sigma)$ is the set of elements in $B_M(\sigma) - B_M^{\cap}(\sigma)$. Finitely many elements of $CB_M(\sigma)$ appearing later in the text may be ignored. We note the following property:

Claim 11. *Suppose $L \in \mathcal{L}$ and* $\text{content}(\sigma) \subseteq L$. *If $B^{\cap}_M(\sigma) \cap (L - \text{content}(\sigma)) = \emptyset$ and $L - C_M(\sigma)$ is finite, then $M^*(\sigma)$ is a grammar for L.*

To see the above claim note that if we construct a sequence τ from σ by inserting elements $x \in [L - (\text{content}(\sigma) \cup C_M(\sigma))]$ after the initial segment σ' of σ such that $x \in C_M(\sigma')$, then $M^*(\sigma) = M^*(\tau)$. Thus, $M^*(\sigma) = M^*(\sigma \diamond T') = M^*(\tau \diamond T')$, where T' is a text for $C_M(\sigma)$. It follows that $M^*(\sigma)$ must be a grammar for L. This completes the proof of Claim 11.

We will now describe our learner N which simulates M. The conjectures of the learner N will be of the form $f(\sigma, m, \alpha)$, where f is a recursive function mapping SEQ $\times\ \mathbb{N} \times$ SEQ to $\mathbb{N}$, and is defined later below.

For testing whether the parameters (σ, m, α) are good (in the sense that the elements of the language $W_{M(\sigma,m,\alpha)}$ which are greater than m do not belong to $CB_M(\sigma)$), we use the following predicate P. Here x can be considered as a time bound for checking.

Let P be such that for all $\sigma \in$ SEQ, $m \in \mathbb{N}$ 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 $m < w \leq x]$.

Now we define the learner N:

$N(\emptyset) = f(\emptyset, 0, \emptyset)$. $N(f(\sigma, m, \alpha), x)$ is defined as follows (that is, for previous conjecture $f(\sigma, m, \alpha)$ and new datum x, 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. Note that $W_{f(\sigma,m,\alpha)}$ does not depend on α; usage of α in the proof is mainly for memorising some mind change points of M.

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 $x \in B_M^\cap(\sigma)$, 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 $\text{content}(\tau) \subseteq A_s$ 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.

(* Intuitively, we would like not to enumerate elements which cause a mind change to ensure cautiousness, weak monotonicity and decisiveness. However, this causes problems in learnability (specially of finite languages). Thus, this step is used to include "safe elements" which are known to be included in "future conjectures" (see cases (ii), (iv) and (v) in the definition of N). This "inclusion" allows learnability of finite languages, as discussed later below, without violating cautiousness, weak monotonicity and decisiveness (see Case 2 in the analysis below). *)

(c) If the condition "there exists an $x \in A_s$ such that $x \in B_M^\cap(\sigma)$" in (a) and the condition "there exists an $x \in A_s$ such that $x > m$, and $[x \in CB_M(\sigma)$ or $P(\sigma, m, x)]$" in (b) both fail, then enumerate A_s, and go to stage $s + 1$.

End stage s

Now let T be a text for $L \in \mathcal{L}$, and let $f(\sigma_n, m_n, \alpha_n)$ be the output of $N^*(T[n])$. By definition of N it is easy to verify that

- $\text{content}(\sigma_n \diamond \alpha_n) \subseteq \text{content}(T[n])$,
- $\sigma_n \subseteq \sigma_{n+1}$,
- $\sigma_n \diamond \alpha_n \subseteq \sigma_{n+1} \diamond \alpha_{n+1}$,
- if $\sigma_n \neq \sigma_{n+1}$, then $\alpha_{n+1} = \emptyset$, and
- if $\sigma_n = \sigma_{n+1}$, then $m_{n+1} \geq m_n$.

Furthermore, note that on input $T(n)$, if the choice in the definition of N is Case (iv) or (vi) then, $M^*(\sigma_n \diamond \alpha_n) = M^*(\sigma_n \diamond \alpha_n \diamond x)$. Using this and above mentioned properties of N, it is easy to verify by induction on n that $M^*(T[n]) = M^*(\sigma_n \diamond \alpha_n)$.

Now, if $\sigma_n \diamond \alpha_n \neq \sigma_{n+1} \diamond \alpha_{n+1}$, then $M^*(T[n]) = M^*(\sigma_n \diamond \alpha_n) \neq M^*(\sigma_{n+1} \diamond \alpha_{n+1}) = M^*(T[n+1])$. Thus, as M^* converges on T, we immediately have that $\lim_{n\to\infty} \sigma_n \diamond \alpha_n$ exists and thus $\lim_{n\to\infty} \sigma_n$ and $\lim_{n\to\infty} \alpha_n$ exist. Let n_0 be large enough so that, for all $n \geq n_0$, $\sigma_n = \sigma_{n_0}$ and $\alpha_n = \alpha_{n_0}$. Note that, for all $n \geq n_0$ (as Case (ii) in the definition of N would not apply), $T(n) \notin B_M^\cap(\sigma_{n_0})$.

For showing that N **TxtItEx**-learns L, we split our analysis based on whether L is finite or infinite.

Suppose L is a finite language. First note by Claim 11 that if $\text{content}(\sigma) \subseteq L$ and $L \cap B_M^\cap(\sigma) = \emptyset$, then $W_{M^*(\sigma)} = L$. Thus for σ and α such that $\text{content}(\sigma) \cup \text{content}(\alpha) \subseteq L$, and $L \cap B_M^\cap(\sigma) = \emptyset$, using reverse induction on the number of mind changes made by M^* on σ (which is bounded by $\text{card}(L)$ due to M being canny), we show that $W_{f(\sigma,m,\alpha)} = L$. To see this, note that for such

σ, in the definition of f, 2(a) never applies. Now, for the base case of induction, if the number of mind changes made by M on σ is card(L), then as M is canny, $CB_M(\sigma) = \emptyset$, and thus in the definition of f step 2(b) also does not apply. Thus, $W_{f(\sigma,m,\alpha)} = W_{M^*(\sigma)} = L$. Inductively, assuming that for all $\sigma' \supseteq \sigma$ such that content$(\sigma') \subseteq L$ and M^* makes more mind changes on σ' than on σ, $W_{f(\sigma',0,\emptyset)} = L$, we get from 2(b) in the definition of f that $W_{f(\sigma,m,\alpha)}$ will enumerate L. It follows that, for all $n \geq n_0$, $W_{f(\sigma_n,m_n,\alpha_n)}$ is a grammar for L. Furthermore, for $n \geq n_0$, m_n is monotonically non-decreasing and m_n cannot be greater than $\max(L)$ (by definition of N). Thus, $\lim_{n\to\infty} m_n$ exists and hence N **TxtItEx**-learns L.

Now, suppose L is an infinite language in $\mathcal{L}$. If $\alpha_{n_0} \neq \emptyset$, then clearly $m = \lim_{n\to\infty} m_n$ also exists (as Case (iv) in the definition of N does not apply for inputs $T(n_0), T(n_0+1), \ldots$). Furthermore, as $B_M^{\cap}(\sigma_{n_0}) \cap L = \emptyset$, by Claim 11 we also have $W_{M^*(\sigma_{n_0})} = L$. If $\alpha_{n_0} = \emptyset$, then as $M^*(T) = M^*(\sigma_{n_0})$, we have that $W_{M^*(\sigma_{n_0})} = L$ and all but finitely many of the elements of L do not belong to $B_M(\sigma_{n_0})$. Thus, in this case also $m = \lim_{n\to\infty} m_n$ exists (as $P(\sigma_n, m_n, T(n))$ does not hold for large enough m_n). In both cases, m bounds all the elements of L which are in $B_M(\sigma_{n_0})$. Thus, $f(\sigma_{n_0}, m, \alpha_{n_0})$ is a grammar for L (as item 2(c) in the definition of f would apply).

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

$W_{f(\sigma,m,\alpha)} \subseteq W_{f(\sigma,m',\alpha')}$ for $m \leq m'$ and all $\alpha, \sigma, \alpha' \in \text{SEQ}$ — (P1)

Thus, $W_{f(\sigma_n,m_n,\alpha_n)} \subseteq W_{f(\sigma_{n+1},m_{n+1},\alpha_{n+1})}$, if Case (iii), (iv) or (vi) applied in the definition of N when input $T(n)$ was considered. Now suppose Case (ii) or (v) is used when N reads input $T(n)$. Thus, $\sigma_{n+1} = \sigma_n \diamond \alpha_n \diamond T(n)$ and either $T(n) \in B_M^{\cap}(\sigma_n)$ or $[T(n) > m_n$ and $(T(n) \in CB_M(\sigma_n)$ or $P(\sigma_n, m_n, T(n)))]$ holds.

Case 1: content$(\alpha_n \diamond T(n)) \not\subseteq W_{f(\sigma_n,m_n,\alpha_n)}$.

In this case clearly content$(T[n+1]) \supseteq$ content$(\sigma_n \diamond \alpha_n \diamond T(n))$ and thus, content$(T[n+1]) \not\subseteq W_{f(\sigma_n,m_n,\alpha_n)}$, so mind change is safe (weak monotonic).

Case 2: content$(\alpha_n \diamond T(n)) \subseteq W_{f(\sigma_n,m_n,\alpha_n)}$.

Let s be least such that content$(\alpha_n \diamond T(n))$ is contained in A_s as in stage s of $W_{f(\sigma_n,m_n,\alpha_n)}$. Then, the definition of $W_{f(\sigma_n,m_n,\alpha_n)}$ step 2(b) ensures that $W_{f(\sigma_n,m_n,\alpha_n)}$ enumerates $A_t, t \geq s$, only if A_t is contained in $W_{f(\sigma_n \diamond \alpha_n \diamond T(n),0,\emptyset)}$ (note that the case of $A_t =$ content(σ_n), already satisfies $A_t \subseteq W_{f(\sigma_n \diamond \alpha_n \diamond T(n),0,\emptyset)}$).

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, for all $n' > n$, $W_{f(\sigma_{n'},m_{n'},\alpha_{n'})}$ contains content$(\sigma_{n'})$, which contains content$(\alpha_n \diamond T(n))$. Thus, N never returns to the conjecture $W_{f(\sigma_n,m_n,\alpha_n)}$, which does not contain content$(\alpha_n \diamond T(n))$. 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. $\square$

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 $x \in \mathrm{content}(T)$, 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 Theorem 15, showing that anything that can be iteratively learnt can be so learnt conservatively or strongly decisively *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 12. 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 a text 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, 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 13. *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 14. $\mathbf{TxtItEx} \subset \mathbf{Txt}^{\mathrm{fat}}\mathbf{ItEx} \subset \mathbf{TxtGEx}$.

Proof. The first inequality is easy to see using the class $\mathcal{L} = \{\{1, 2, 3, \ldots\}\} \cup \{L \mid 0 \in L, L \text{ is finite}\}$. Clearly, $\mathcal{L}$ is in $\mathbf{Txt}^{\mathrm{fat}}\mathbf{ItEx}$, as the learner can just output a grammar for $\{1, 2, 3, \ldots\}$, until it sees the element 0. From then on, the learner can just output the set of elements seen in the input after having seen the element 0. It is well-known [CCJS07] that $\mathcal{L} \notin \mathbf{TxtItEx}$.

Regarding the second inequality, let L and H be a recursively inseparable pair of languages [Rog67]. Consider the class $\mathcal{L} = \{L, \mathbb{N}\} \cup \{L \cup \{x\} \mid x \in H\}$. $\mathcal{L}$ 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 mind change 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 similar 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 [BB75,Ful90] for M on $L \cup \{x\}$, and thus M does not $\mathbf{Txt}^{\text{fat}}\mathbf{ItEx}$-learn the language $L \cup \{x\}$. Thus, M cannot $\mathbf{Txt}^{\text{fat}}\mathbf{ItEx}$-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 computational limitations which cannot be compensated by having fat text. Next we show that fat text always allows for learning conservatively (as well as strongly decisively) for iterative learners.

Theorem 15. $\mathbf{Txt}^{\text{fat}}\mathbf{ItEx} = \mathbf{Txt}^{\text{fat}}\mathbf{ItConsvEx} = \mathbf{Txt}^{\text{fat}}\mathbf{ItSDecEx}$.

Proof. Note that an iterative learner learning from fat text is sink-locking on fat texts: Whenever the learner M learns L and there is for the current hypothesis e an x with $M(e, x) \neq e$ then this x will eventually appear in the fat text and the learner will eventually make a mind change; thus the learner only converges to a hypothesis e if $M(e, x) = e$ for all $x \in L$. Thus one can apply Theorem 8 and sees that the learner N constructed there learns, from fat texts, the same languages as the original learner M and is, in addition, conservative and weakly monotonic. Furthermore, one can follow the arguments there to see that the learner can be made strongly decisive. □

The following proposition follows from Lemma 13 and Theorems 2 and 3.

Proposition 16. **(a)** *There exists a class of languages which is* $\mathbf{TxtItMonEx}$, $\mathbf{TxtItSDecEx}$, $\mathbf{TxtItConsvEx}$*-learnable but not* $\mathbf{Txt}^{\text{fat}}\mathbf{SMonEx}$*-learnable.*
(b) *There is a class which is* $\mathbf{TxtItSDecEx}$*-learnable (and therefore also* $\mathbf{TxtItConsvEx}$*-learnable) but not* $\mathbf{Txt}^{\text{fat}}\mathbf{ItMonEx}$ *or* $\mathbf{Txt}^{\text{one-one}}\mathbf{ItMonEx}$*-learnable.*

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

Theorem 17. $\mathbf{TxtItSMonEx} \not\subseteq \mathbf{Txt}^{\text{fat}}\mathbf{ItSNUShEx}$.

We next show that learning from 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 18 (Kötzing [Köt14]). For all $p \in \mathcal{R}$, we let

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

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

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 19. *Let δ be a pseudo-semantic restriction allowing for simulation on consistent text (see Definition 12). 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 20. *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 $L_{e,z} = \{(e, x, y) \mid x = z$ 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 $\{(e,x,y) \mid x,y \in \mathbb{N}\}$ 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 21. *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:

$$\begin{aligned} L_0 &= \{2m \mid m \in \mathbb{N}\}, \\ L_{2n+1} &= \{2m \mid m \in \mathbb{N}, m \leq n\} \cup \{2n+1\} \text{ and} \\ L_{2n+2} &= \{2m \mid m \in \mathbb{N}, m \leq n+1\} \cup \{2n+1\}. \end{aligned}$$

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 $\text{content}(\sigma) = \{2m \mid m \leq n\}$ for some n. Now, $M^*(\sigma \diamond 2n+2 \diamond 2n+1 \diamond \#^r) = M^*(\sigma \diamond 2n+1 \diamond \#^r)$, for all r, and thus M fails to identify at least one of L_{2n+1} and L_{2n+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 $\mathcal{L}$) [LZ93] iff $\{H_e \mid e \in \mathbb{N}\} = \mathcal{L}$. 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 22. *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 $x \in H_e \cup \{\#\}$, $M(e,x) = e$.*

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 $x \in H_e \cup \{\#\}$. □

The next 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 23. *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. 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 $\{\langle e, 2x\rangle \mid x \in \mathbb{N}\}$ and L_{2e+1} contain $\{\langle e, 2x+1\rangle \mid x \in \mathbb{N}\}$.
2. Search for $\sigma_{2e}, \sigma_{2e+1}$ such that
   - content$(\sigma_{2e}) \subseteq \{\langle e, 2x\rangle \mid x \in \mathbb{N}\}$ and
   - content$(\sigma_{2e+1}) \subseteq \{\langle e, 2x+1\rangle \mid x \in \mathbb{N}\}$ 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 $M_e^*(\sigma_{2e}\diamond\langle e, x_e\rangle) \neq M_e^*(\sigma_{2e+1}\diamond \langle e, x_e\rangle)$.
4. If and when such an $\langle e, x_e\rangle$ is found, enumerate $\langle e, x_e\rangle$ 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 **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 2 in the construction of L_{2e} and L_{2e+1} must succeed, as otherwise, M_e does not identify at least one of $L_{2e} = \{\langle e, 2x\rangle \mid x \in \mathbb{N}\}$ or $L_{2e+1} = \{\langle e, 2x+1\rangle \mid x \in \mathbb{N}\}$. 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 $M_e^*(\sigma_{2e+1})$ is a grammar (in hypothesis space $\mathcal{H}^e$) for L_{2e} and the other one is a grammar for L_{2e+1}. Now, step 3 must also succeed to find $\langle e, x_e\rangle$ as both L_{2e} and L_{2e+1} contain content$(\sigma_{2e}) \cup$ content(σ_{2e+1}). But then, both L_{2e} and L_{2e+1} contain $\langle e, x_e\rangle$ and thus M_e violates Lemma 22.

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_e in hypothesis space $\mathcal{H} = (H_i)_{i\in\mathbb{N}}$.

On any input element $\langle e, x\rangle$, 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 h_{2e} infinitely often and h_{2e+1} only finitely often (respectively, h_{2e+1} infinitely often and h_{2e} only finitely often). Thus, M $\mathbf{Txt}^{\text{one-one}}\mathbf{ItEx}$-learns $\mathcal{L}$ using hypothesis space $\mathcal{H}$. □

Theorem 2 and Theorem 3 witness that the hierarchy $\mathbf{SMonEx} \subseteq \mathbf{MonEx} \subseteq \mathbf{WMonEx}$ holds for iterative learners. It is easy to see that one can use one-one class preserving hypothesis spaces for the learners in these theorems.

We now consider learning by *reliable* learners. A learner is *reliable* (see [BB75,Min76]) 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 $= \{L \mid L \text{ is finite}\}$.

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 22). 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 $\mathcal{L} = \{L \mid 2 \leq \text{card}(L) \leq 3\}$. 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 for any σ and τ containing at most three distinct elements, it is not possible that content(σ) $\neq$ content(τ) and $M^*(\sigma) = M^*(\tau)$. To see this, note that if both σ and τ contain at least two distinct elements, then $M^*(\sigma \diamond \#^\infty) = M^*(\tau \diamond \#^\infty)$, but $\sigma \diamond \#^\infty$ and $\tau \diamond \#^\infty$ are texts for different languages in $\mathcal{L}$. If at least one of σ and τ contains at most one distinct element, then let w, z be such that content(σ) $\cup \{w, z\} \neq$ content(τ) $\cup \{w, z\}$ and card(content(σ) $\cup \{w, z\}) \leq 3$ and card(content(τ) $\cup \{w, z\}) \leq 3$. Note that there exist such w, z (if τ contains at least as many elements as σ, and $y \in$ content(τ) $-$ content(σ), then we can choose w, z to be different from y such that card(content(τ) $\cup \{w, z\}) \leq 3$). Now, $M^*(\sigma \diamond (w \diamond z)^\infty) = M^*(\tau \diamond (w \diamond z)^\infty)$ even though $\sigma \diamond (w \diamond z)^\infty$ and $\tau \diamond (w \diamond z)^\infty$ are texts for different languages in $\mathcal{L}$.

In particular we may assume without loss of 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 22, 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 22 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 22 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. □

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 $X = \{e \mid D_e \in \mathcal{L}\}$ 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 $\mathcal{L} \subseteq \text{FIN}$ is nonempty and the set X of its canonical indices is 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 X_s denote the elements enumerated into X within s steps; if $e \in X_s$ then $H_{\langle e,s \rangle} = D_e$ else $H_{\langle e,s \rangle}$ is some default finite set in $\mathcal{L}$.

The iterative learner always conjectures indices of the form $\langle e, s \rangle$ where e is maintained such that D_e contains exactly 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 X and which either grows or stabilises. The initial hypothesis of the iterative learner 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 satisfying $D_e = D_d \cup \{x\} - \{\#\}$;
- If $e \in X_s$ 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 exactly 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 learnt is finite, from some time on the parameter e will stabilise at the correct value. In the case that $e \notin X$ 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 $e \in X$ the parameter s will stop growing once $e \in X_s$ 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 $e \in X_s$; 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. $\square$

7 Syntactic versus Semantic Conservativeness

A learner M is called *semantically conservative* iff whenever $W_{M^*(\sigma)} \neq W_{M^*(\sigma\diamond\tau)}$ then content$(\sigma \diamond \tau) \not\subseteq W_{M^*(\sigma)}$. 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 $\sigma_n \diamond 3h \diamond 3m + 2 \diamond \tau_n$ and $\sigma_n \diamond 3h \diamond 3m + 2 \diamond \tau_n \diamond 0$ the same hypothesis e_n while it outputs a different index on $\sigma_n \diamond 3h \diamond 3m + 2 \diamond \tau_n \diamond 3h$. 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 L_{3n}.

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 case) 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 for $\mathcal{L}$ can work as follows. It conjectures grammar for $\emptyset$ until either the number $3n+1$ or $3m+2$ are observed in the input (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 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 $\mathrm{content}(\sigma_n) \cup \{3m+2\}$ then the learner pads this number into the previous hypothesis without making a semantic mind change. If it sees a further number $3k'$ outside $\mathrm{content}(\sigma_n) \cup \{3k, 3m+2\}$ 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. 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 $\mathrm{content}(\sigma_n) \cup \{3m+2\}$ and therefore also the second semantic mind change is semantically conservative. □

8 Conclusion

We considered iterative learning in the limit and gave a complete map of various update constraints in this setting (Figure 1). In particular, we showed that even decisive learning does not reduce learning power below unconstrained learning, but strongly decisive learning does.

However, as the proofs have shown, many intricate tricks to fulfill these requirements had to be employed which might not be possible in more applied settings (for example it was heavily exploited that W-indices were used as hypotheses and not, for instance, regular expressions for learning a subclass of regular languages). While it was important to characterize the relations of the learning criteria in this very general setting, further work should address other, more specific settings. As a first step we gave some results concerning class preserving hypothesis spaces in Section 6, but clearly more work can be done towards the end of understanding the impact of the hypothesis space used in iterative learning.

Acknowledgements. A preliminary version of this paper appeared at the conference ALT 2014 [JKMS14]. We thank the referees of the journal and of ALT 2014 for several helpful comments.

References

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

[Bar74] J. Bārzdiņš. Inductive inference of automata, functions and programs. In *Proceedings of the 20th International Congress of Mathematicians, Vancouver*, pages 455–460, 1974. In Russian. English translation in American Mathematical Society Translations: Series 2, 109:107-112, 1977.

[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.

[CCJS07] Lorenzo Carlucci, John Case, Sanjay Jain and Frank Stephan. Results on Memory-Limited U-Shaped Learning. *Information and Computation*, 205:1551–1573, 2007.

[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 the 23rd International Conference on Computational Learning Theory*, COLT 2010, Proceedings. Pages 181–193, Omnipress 2010.

[CL82] John Case and Chris Lynes. Machine inductive inference and language identification. *Proceedings of the 9th International Colloquium on Automata, Languages and Programming*, ICALP 1982. Springer LNCS 140:107–115, 1982.

[CM08a] John Case and Samuel E. Moelius. Optimal language learning. *Proceedings of the 19th International Conference on Algorithmic Learning Theory*, ALT 2008. Springer LNAI 5254:419–433, 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.

[GJS13] Ziyuan Gao, Sanjay Jain and Frank Stephan. On conservative learning of recursively enumerable languages. *Proceedings of the 9th Conference on Computability in Europe: The Nature of Computation, Logic, Algorithms, Applications*, CiE 2013. Springer LNCS 7921:181–190, 2013.

[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. *New Generation Computing*, 8:349–360, 1991.

[JKMS14] Sanjay Jain, Timo Kötzing, Junqi Ma and Frank Stephan. On the role of update constraints and text-types in iterative learning. *Proceedings of the 25th Inter-*

national Conference on Algorithmic Learning Theory, ALT 2014. Springer LNAI 8776:55–69, 2014.

[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. *Proceedings of the 31st International Symposium on Theoretical Aspects of Computer Science*, STACS 2014. LIPIcs 25, 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 the 2nd International Workshop on Nonmonotonic and Inductive Logic*. Springer LNAI 659:254–269, 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.

[Min76] E. Minicozzi, Some natural properties of strong identification in inductive inference, *Theoretical Computer Science*, 2:345–360, 1976.

[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.

[Wie76] R. Wiehagen. Limes-Erkennung rekursiver Funktionen durch spezielle Strategien. *Journal of Information Processing and Cybernetics (EIK)*, 12(1–2):93–99, 1976.

[Wie90] Rolf Wiehagen. A thesis in inductive inference. *Proceedings of the 1st Workshop on Nonmonotonic and Inductive Logic*. Springer LNCS 543:184–207, 1990.