Theoretical Computer Science 650 (2016) 4–24

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

A map of update constraints in inductive inference

Timo Kötzing [a,*], Raphaela Palenta [b]

[a] Friedrich-Schiller-Universität Jena, Germany
[b] Technische Universität Müchen, Germany

ARTICLE INFO

Article history:
Available online 29 July 2016

Keywords:
Inductive inference
Language learning
Update constraints

ABSTRACT

We investigate how different learning restrictions reduce learning power and how the different restrictions relate to one another. We give a complete map for nine different restrictions both for the cases of complete information learning and set-driven learning. This completes the picture for these well-studied *delayable* learning restrictions. A further insight is gained by different characterizations of *conservative* learning in terms of variants of *cautious* learning.

Our analyses greatly benefit from general theorems we give, for example showing that learners which have to obey only delayable restrictions can always be assumed total.

1. Introduction

This paper is set in the framework of *inductive inference*, a branch of (algorithmic) learning theory. This branch analyzes the problem of algorithmically learning a description for a formal language (a computably enumerable subset of the set of natural numbers) when presented successively all and only the elements of that language. For example, a learner h might be presented more and more even numbers. After each new number, h outputs a description for a language as its conjecture. The learner h 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 h sees an even number not divisible by 4, it might change this guess to a program for the set of all multiples of 2.

Many criteria for deciding whether a learner h is *successful* on a language L have been proposed in the literature. Gold, in his seminal paper [9], gave a first, simple learning criterion, **TxtGEx**-*learning*,[1] 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 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 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 [11]).

A wealth of learning criteria can be derived from **TxtGEx**-learning by adding restrictions on the intermediate conjectures and how they should relate to each other and the data. For example, one could require that a conjecture which is consistent with the data must not be changed; this is known as *conservative* learning and known to restrict what classes of languages

* Corresponding author.
E-mail address: timo.koetzing@hpi.de (T. Kötzing).

[1] **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*.

http://dx.doi.org/10.1016/j.tcs.2016.07.028
0304-3975/

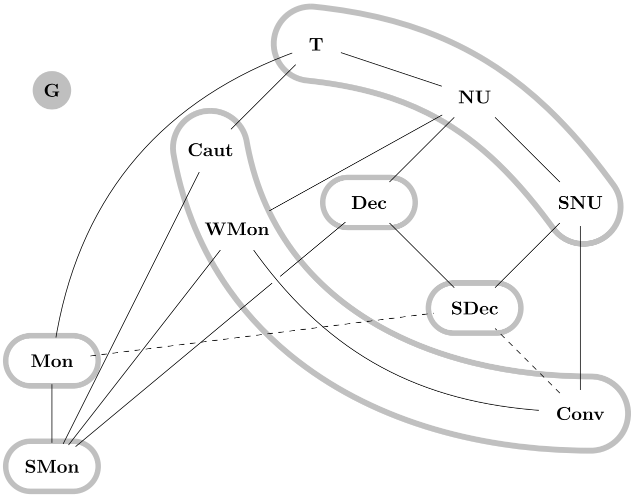


Fig. 1. Relation of criteria.

can be learned ([1], we use **Conv** to denote the restriction of conservative learning). Additionally to conservative learning, the following learning restrictions are considered in this paper (see Section 2.1 for a formal definition of learning criteria including these learning restrictions).

In *cautious* learning (**Caut**, [18]) the learner is not allowed to ever give a conjecture for a strict subset of a previously conjectured set. In *non-U-shaped* learning (**NU**, [3]) a learner may never *semantically* abandon a correct conjecture; in *strongly non-U-shaped* learning (**SNU**, [7]) not even syntactic changes are allowed after giving a correct conjecture.

In *decisive* learning (**Dec**, [18]), a learner may never (semantically) return to a *semantically* abandoned conjecture; in *strongly decisive* learning (**SDec**, [14]) the learner may not even (semantically) return to *syntactically* abandoned conjectures. Finally, a number of monotonicity requirements are studied [10,24,17]: in *strongly monotone* learning (**SMon**) the conjectured sets may only grow; in *monotone* learning (**Mon**) only incorrect data may be removed; and in *weakly monotone* learning (**WMon**) the conjectured set may only grow while it is consistent.

The main question is now whether and how these different restrictions reduce learning power. For example, non-U-shaped learning is known not to restrict the learning power [3], and the same for strongly non-U-shaped learning [7]; on the other hand, decisive learning *is* restrictive [3]. The relations of the different monotone learning restriction were given in [17]. Conservativeness is long known to restrict learning power [1], but also known to be equivalent to weakly monotone learning [16,12].

Cautious learning was shown to be a restriction but not when added to conservativeness in [18,19], similarly the relationship between decisive and conservative learning was given. In Exercise 4.5.4B of [19] it is claimed (without proof) that cautious learners cannot be made conservative; we claim the opposite in Theorem 4.4.

This list of previously known results leaves a number of relations between the learning criteria open, even when adding trivial inclusion results (we call an inclusion trivial iff it follows straight from the definition of the restriction without considering the learning model, for example strongly decisive learning is included in decisive learning; formally, trivial inclusion is inclusion on the level of learning restrictions as predicates, see Section 2.1). With this paper we now give the complete picture of these learning restrictions. The result is shown as a map in Fig. 1. A solid black line indicates a trivial inclusion (the lower criterion is included in the higher); a dashed black line indicates an inclusion which is not trivial. A gray box around criteria indicates equality of (learning of) these criteria.

A different way of depicting the same results is given in Fig. 2 (where solid lines indicate inclusion). Results involving monotone learning can be found in Section 7, results on the particularly difficult relations of decisive learning in Section 5, all others in Section 4.

For the important restriction of conservative learning we give the characterization of being equivalent to cautious learning. Furthermore, we show that even two weak versions of cautiousness are equivalent to conservative learning. Recall that cautiousness forbids to return to a strict subset of a previously conjectured set. If we now weaken this restriction to forbid to return to *finite* subsets of a previously conjectured set we get a restriction still equivalent to conservative learning. If we forbid to go down to a correct conjecture, effectively forbidding to ever conjecture a superset of the target language, we also obtain a restriction equivalent to conservative learning. On the other hand, if we weaken it so as to only forbid going to *infinite* subsets of previously conjectured sets, we obtain a restriction equivalent to no restriction. These results can be found in Section 4.

In *set-driven* learning [23] the learner does not get the full information about what data has been presented in what order and multiplicity; instead, the learner only gets the set of data presented so far. For this learning model it is known that, surprisingly, conservative learning is no restriction [16]! We complete the picture for set driven learning by showing that

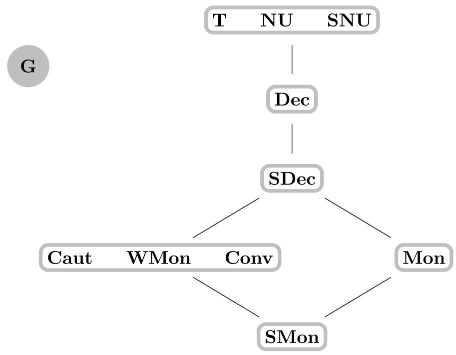


Fig. 2. Partial order of delayable learning restrictions in Gold-style learning.

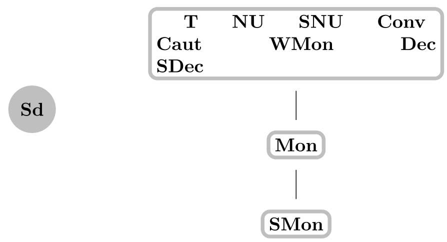


Fig. 3. Hierarchy of delayable learning restrictions in set-driven learning.

set-driven learners can always be assumed conservative, strongly decisive and cautious, and by showing that the hierarchy of monotone and strongly monotone learning also holds for set-driven learning. The situation is depicted in Fig. 3. These results can be found in Section 6.

1.1. Techniques

A major emphasis of this paper is on the techniques used to get our results. These techniques include specific techniques for specific problems, as well as general theorems which are applicable in many different settings. The general techniques are given in Section 3, one main general result is as follows. It is well-known that any **TxtGEx**-learner h learning a language L has a *locking sequence*, a sequence σ of data from L such that, for any further data from L, the conjecture does not change and is correct. However, there might be texts such that no initial sequence of the text is a locking sequence. We call a learner such that any text for a target language contains a locking sequence *strongly locking*, a property which is very handy to have in many proofs. Fulk [8] showed that, without loss of generality, a **TxtGEx**-learner can be assumed strongly locking, as well as having many other useful properties (we call this the *Fulk normal form*, see Definition 3.8). For many learning criteria considered in this paper it might be too much to hope for that they allow for learning by a learner in Fulk normal form. However, we show in Corollary 3.7 that we can get a weaker kind of normal form for many learning criteria: the learners can be assumed strongly locking, total, and what we call *syntactically decisive*, never *syntactically* returning to syntactically abandoned hypotheses.

The main technique we use to show that something is decisively learnable, for example in Theorem 7.3, is what we call *poisoning* of conjectures. In the proof of Theorem 7.3 we show that a class of languages is decisively learnable by simulating a given monotone learner h, but changing conjectures as follows. Given a conjecture e made by h, if there is no mind change in the future with data from conjecture e, the new conjecture is equivalent to e; otherwise it is suitably changed, *poisoned*, to make sure that the resulting learner is decisive. This technique was also used in [6] to show strongly non-U-shaped learnability.

Finally, for showing classes of languages to be not (strongly) decisively learnable, we adapt a technique known in computability theory as a "priority argument" (note, though, that we do not deal with oracle computations). We use this technique to reprove that decisiveness is a restriction to **TxtGEx**-learning (as shown in [3]). Note that this proof is based on the same ideas as the proof in [3], but rephrased in terms of a priority argument. We use this rephrased proof as the starting point for a variation with which we show that strongly decisive learning is a restriction to decisive learning.

A previous version of this paper appeared in the proceedings of ALT'14 [15].

2. Mathematical preliminaries

Unintroduced notation follows [21], a textbook on computability theory.

$\mathbb{N}$ denotes the set of natural numbers, $\{0, 1, 2, \ldots\}$. The symbols $\subseteq$, $\subset$, $\supseteq$, $\supset$ respectively denote the subset, proper subset, superset and proper superset relation between sets; $\setminus$ denotes set difference. $\emptyset$ and λ denote the empty set and the empty sequence, respectively. The quantifier $\forall^\infty x$ means "for all but finitely many x". With dom and range we denote, respectively, domain and range of a given function.

We let $\langle\cdot,\cdot\rangle$ be a linear time computable, linear time invertible, pairing function [20] (a pairing function is a 1–1 and onto mapping $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$). 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. For finite sequences, we suppose that for any $\sigma \subseteq \tau$ we have that the code number of σ is at most the code number of τ. We let $\mathbb{S}\mathrm{eq}$ denote the set of all (finite) sequences, and $\mathbb{S}\mathrm{eq}_{\leq t}$ the (finite) set of all sequences of length at most t using only elements $\leq t$. We let $\diamond$ denote concatenation on sequences. When σ is a non-empty sequence, we let σ^- be the sequence where the last element of σ is removed. We assume a well-ordering on all sequences which is compatible with subsequences; similarly for all finite sets.

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 $f(x)\downarrow = p$. We let $\mathfrak{P}$ denote the set of all partial functions $\mathbb{N} \to \mathbb{N}$ and $\mathfrak{R}$ the set of all total such functions.

$\mathcal{P}$ and $\mathcal{R}$ denote, respectively, the set of all partial computable and the set of all total computable functions (mapping $\mathbb{N} \to \mathbb{N}$).

We let φ be any fixed acceptable programming system for $\mathcal{P}$ (an acceptable programming system could, for example, be based on a natural programming language such as C or Java, or on Turing machines). Further, we let φ_p denote the partial computable function computed by the φ-program with code number p. A set $L \subseteq \mathbb{N}$ is *computably enumerable (ce)* iff it is the domain of a computable function. Let $\mathcal{E}$ denote the set of all ce sets. We let W be the mapping such that $\forall e : W(e) = \mathrm{dom}(\varphi_e)$. For each e, we write W_e instead of $W(e)$. 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.

We let Φ be a Blum complexity measure associated with φ (for example, for each e and x, $\Phi_e(x)$ could denote the number of steps that program e takes on input x before terminating). For all e and t we let $W_e^t = \{x \leq t \mid \Phi_e(x) \leq t\}$ (note that a complete description for the finite set W_e^t is computable from e and t). The symbol # is pronounced *pause* and is used to symbolize "no new input data" in a 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 computable functions. For any function T and all i, we use $T[i]$ to denote the sequence $T(0), \ldots, T(i-1)$ (the empty sequence if $i = 0$ and undefined, if any of these values is undefined).

We will use the existence of a 1–1 *padding function* $\mathrm{pad} \in \mathcal{R}$ such that

$$\forall e, i : \varphi_{\mathrm{pad}(e,i)} = \varphi_e.$$

We will use Case's *Operator Recursion Theorem* (**ORT**), providing *infinitary* self-and-other program reference [4,5,11]. **ORT** itself states that, for all operators Θ there are f with $\forall z : \Theta(\varphi_z) = \varphi_{f(z)}$ and $e \in \mathcal{R}$,

$$\forall a, b : \varphi_{e(a)}(b) = \Theta(e)(a, b).$$

2.1. Learning criteria

In this section we formally introduce our setting of learning in the limit and associated learning criteria. We follow [13] in its "building-blocks" approach for defining learning criteria.

A *learner* is a partial computable function $h \in \mathcal{P}$. A *language* is a ce 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.

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

We define the interaction operators **G**, **Psd** (partially set-driven learning, [22]) and **Sd** (set-driven learning, [23]) as follows. For all learners h, texts T and all i,

$$\mathbf{G}(h, T)(i) = h(T[i]);$$

$$\mathbf{Psd}(h, T)(i) = h(\mathrm{content}(T[i]), i);$$

$$\mathbf{Sd}(h, T)(i) = h(\mathrm{content}(T[i])).$$

Thus, in set-driven learning, the learner has access to the set of all previous data, but not to the sequence as in **G**-learning. In partially set-driven learning, the learner has the set of data and the length of the input sequence.

Successful learning requires the learner to observe certain restrictions, for example convergence to a correct index. These restrictions are formalized 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**, [9]) defined such that, for all sequences of hypotheses p and all texts T,

$$\mathbf{Conv}(p,T) \Leftrightarrow [\forall i : \mathrm{content}(T[i+1]) \subseteq W_{p(i)} \Rightarrow p(i) = p(i+1)];$$

$$\mathbf{Caut}(p,T) \Leftrightarrow [\forall i,j : W_{p(i)} \subset W_{p(j)} \Rightarrow i < j];$$

$$\mathbf{NU}(p,T) \Leftrightarrow [\forall i,j,k : i \le j \le k \ \wedge\ W_{p(i)} = W_{p(k)} = \mathrm{content}(T) \Rightarrow W_{p(j)} = W_{p(i)}];$$

$$\mathbf{Dec}(p,T) \Leftrightarrow [\forall i,j,k : i \le j \le k \ \wedge\ W_{p(i)} = W_{p(k)} \Rightarrow W_{p(j)} = W_{p(i)}];$$

$$\mathbf{SNU}(p,T) \Leftrightarrow [\forall i,j,k : i \le j \le k \ \wedge\ W_{p(i)} = W_{p(k)} = \mathrm{content}(T) \Rightarrow p(j) = p(i)];$$

$$\mathbf{SDec}(p,T) \Leftrightarrow [\forall i,j,k : i \le j \le k \ \wedge\ W_{p(i)} = W_{p(k)} \Rightarrow p(j) = p(i)];$$

$$\mathbf{SMon}(p,T) \Leftrightarrow [\forall i,j : i < j \Rightarrow W_{p(i)} \subseteq W_{p(j)}];$$

$$\mathbf{Mon}(p,T) \Leftrightarrow [\forall i,j : i < j \Rightarrow W_{p(i)} \cap \mathrm{content}(T) \subseteq W_{p(j)} \cap \mathrm{content}(T)];$$

$$\mathbf{WMon}(p,T) \Leftrightarrow [\forall i,j : i < j \wedge \mathrm{content}(T[j]) \subseteq W_{p(i)} \Rightarrow W_{p(i)} \subseteq W_{p(j)}].$$

Fig. 4. Definitions of learning restrictions.

$$\mathbf{Ex}(p,T) \Leftrightarrow p \text{ total } \wedge [\exists n_0 \forall n \ge n_0 : p(n) = p(n_0) \wedge W_{p(n_0)} = \mathrm{content}(T)].$$

Furthermore, we formally define the restrictions discussed in Section 1 in Fig. 4 (where we implicitly require the learning sequence p to be total, as in **Ex**-learning).

A variant on decisiveness is *syntactic decisiveness*, **SynDec**, a technically useful property defined as follows.

$$\mathbf{SynDec}(p,T) \Leftrightarrow [\forall i,j,k : i \le j \le k \ \wedge\ p(i) = p(k) \Rightarrow p(j) = p(i)].$$

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 Fig. 4 are meant to be always used together with **Ex**). With **T** we denote the always true sequence acceptance criterion (no restriction on learning).

A *learning criterion* is a tuple $(\mathcal{C}, \beta, \delta)$, where $\mathcal{C}$ is a set of learners (the admissible learners), β is an interaction operator and δ is a learning restriction; we usually write $\mathcal{C}\mathbf{Txt}\beta\delta$ to denote the learning criterion, omitting $\mathcal{C}$ in case of $\mathcal{C} = \mathcal{P}$. We say that a learner $h \in \mathcal{C}$ $\mathcal{C}\mathbf{Txt}\beta\delta$*-learns* a language L iff, for all texts T for L, $\delta(\beta(h,T),T)$. The set of languages $\mathcal{C}\mathbf{Txt}\beta\delta$-learned by $h \in \mathcal{C}$ is denoted by $\mathcal{C}\mathbf{Txt}\beta\delta(h)$. We write $[\mathcal{C}\mathbf{Txt}\beta\delta]$ to denote the set of all $\mathcal{C}\mathbf{Txt}\beta\delta$-learnable classes (learnable by some learner in $\mathcal{C}$).

3. Delayable learning restrictions

In this section we present technically useful results which show that learners can always be assumed to be in some normal form. We will later always assume our learners to be in the normal form established by Corollary 3.7, the main result of this section.

We start with the definition of *delayable*. Intuitively, a learning criterion δ is delayable iff the output of a hypothesis can be arbitrarily (but not indefinitely) delayed.

Definition 3.1. Let $\vec{R}$ be the set of all non-decreasing $r : \mathbb{N} \to \mathbb{N}$ with infinite limit inferior, i.e. for all m we have $\forall^\infty n : r(n) \ge m$.

A learning restriction δ is *delayable* iff, for all texts T and T' with $\mathrm{content}(T) = \mathrm{content}(T')$, all p and all $r \in \vec{R}$, if $(p,T) \in \delta$ and $\forall n : \mathrm{content}(T[r(n)]) \subseteq \mathrm{content}(T'[n])$, then $(p \circ r, T') \in \delta$. Intuitively, as long as the learner has at least as much data as was used for a given conjecture, then the conjecture is permissible. Note that this condition holds for $T = T'$ if $\forall n : r(n) \le n$.

Note that the intersection of two delayable learning criteria is again delayable and that *all* learning restrictions considered in this paper are delayable.

As the name suggests, we can apply *delaying tricks* (tricks which delay updates of the conjecture) in order to achieve fast computation times in each iteration (but of course in the limit we still spend an infinite amount of time). This gives us equally powerful but total learners, as shown in the next theorem. While it is well-known that, for many learning criteria, the learner can be assumed total, this theorem explicitly formalizes conditions under which totality can be assumed (note that there are also natural learning criteria where totality cannot be assumed, such as consistent learning [11]).

Theorem 3.2. *For any delayable learning restriction δ, we have $[\mathbf{TxtG}\delta] = [\mathcal{R}\mathbf{TxtG}\delta]$.*

Proof. Let h be a $\mathbf{TxtG}\delta$-learner and e such that $\varphi_e = h$. We define a function M such that, for all σ,

$$M(\sigma) = \{\sigma' \subseteq \sigma \mid \Phi_e(\sigma') \le |\sigma|\} \cup \{\lambda\}.$$

We let h' be the learner such that, for all σ,

$$h'(\sigma) = h(\max(M(\sigma)).$$

As h is required to have only total learning sequences, we have that $h(\lambda)\downarrow$; thus, h' is total computable using that M is total computable. Let $\mathcal{L} = \mathbf{TxtG}\delta(h)$, $L \in \mathcal{L}$ and let T be a text for L. Let $r(n) = |\max(M(T[n]))|$. Then we have, for all n, $h'(T[n]) = h(T[r(n)])$. Thus, if we show that $r \in \vec{R}$ we get that h' $\mathbf{TxtG}\delta$-learns L from T using δ delayable. From the definition of M we get that r is non-decreasing and, for all n, $r(n) \leq n$. For any given m there are n, n' with $n' \geq n \geq m$ such that $\Phi_e(T[n]) \leq n'$. Thus, we have $r(n') \geq m$ and, as r is non-decreasing, we get $\forall^\infty n : r(n) \geq m$ as desired. □

Next we define another useful property, which can always be assumed for delayable learning restrictions.

Definition 3.3. A *locking sequence for a learner h on a language L* is any finite sequence σ of elements from L such that $h(\sigma)$ is a correct hypothesis for L and, for sequences τ with elements from L, $h(\sigma \diamond \tau) = h(\sigma)$ [2]. It is well known that every **TxtGEx**-learner h learning a language L has a locking sequence on L. We say that a learning criterion I *allows for strongly locking learning* iff, for each I-learnable class of languages $\mathcal{L}$ there is a learner h such that h I-learns $\mathcal{L}$ and, for each $L \in \mathcal{L}$ and any text T for L, there is an n such that $T[n]$ is a locking sequence of h on L (we call such a learner h *strongly locking*).

With this definition we can give the following theorem.

Theorem 3.4. *Let δ be a delayable learning criterion. Then $\mathcal{R}\mathbf{TxtG}\delta\mathbf{Ex}$ allows for strongly locking learning.*

Proof. Let $\mathcal{L}$ and $h \in \mathcal{R}$ be such that h $\mathcal{R}\mathbf{TxtG}\delta\mathbf{Ex}$-learns $\mathcal{L}$. We define a set $M(\rho, \sigma)$, for all ρ and σ such that

$$M(\rho, \sigma) = \{\tau \mid \mathrm{content}(\tau) \subseteq \mathrm{content}(\sigma) \wedge |\tau| \leq |\sigma| \wedge h(\rho \diamond \tau) \neq h(\rho)\}.$$

Thus, M contains sequences with elements from content(σ) such that h makes a mind change on ρ extended with such a sequence. Additionally, we define a function f recursively such that, for all σ, x and T,

$$f(\lambda) = \lambda;$$

$$f(\sigma \diamond x) = \begin{cases} f(\sigma), & \text{if } M(f(\sigma), \sigma \diamond x) = \emptyset; \\ f(\sigma) \diamond \min(M(f(\sigma), \sigma \diamond x)) \diamond \sigma \diamond x, & \text{otherwise}; \end{cases}$$

$$f(T) = \lim_{n \to \infty} f(T[n]).$$

Intuitively, f searches for longer and longer sequences which are *not* locking sequences. We let h' be the learner such that, for all σ,

$$h'(\sigma) = h(f(\sigma)).$$

Note that f is total (as h is total), and thus h' is total.

Let $L \in \mathcal{L}$ and T be a text for L. We will show now that $f(T)$ converges to a finite sequence.

Claim 1. *We have that $f(T)$ is finite.*

Proof of Claim 1. By way of contradiction, suppose that $f(T)$ is infinite, and let $T' = f(T)$. As $f(T)$ is infinite we get, for every n, an $m > n$ such that $f(T[m]) \neq f(T[n])$. Then we have

$$\mathrm{content}(T[n]) \subseteq \mathrm{content}(f(T[m])).$$

As this holds for every n, we get content$(T) \subseteq$ content$(f(T))$. From the construction of f we know that content$(f(T)) \subseteq$ content(T). Thus, $f(T)$ is a text for L. From the construction of M we get that h does not **TxtGEx**-learns L from T' as h changes infinitely often its mind, a contradiction. □ (for Claim 1)

Next, we will show that h' converges on T and h' is strongly locking. As $f(T)$ is finite, there is n_0 such that, for all $n \geq n_0$,

$$f(T[n]) = f(T[n_0]).$$

As $f(T)$ converges to $f(T[n_0])$, we get from the construction of M that $f(T[n_0])$ is a locking sequence of h on L. Therefore we get that, for all $\tau \in \mathbb{S}\mathrm{eq}(L)$,

$$f(T[n_0]) = f(T[n_0] \diamond \tau)$$

and therefore

$$h'(T[n_0]) = h'(T[n_0] \diamond \tau).$$

Thus, h' is strongly locking and converges on T.

To show that h' fulfills the δ-restriction, we let $T' = f(T[n_0]) \diamond T$ be a text for L starting with $f(T[n_0])$. Let r be such that

$$r(n) = \begin{cases} |f(T[n])|, & \text{if } n \le n_0; \\ r(n_0) + n - n_0, & \text{otherwise.} \end{cases}$$

We now show

$$h(T'[r(n)]) = h'(T[n]).$$

Case 1: $n \le n_0$. Then we get

$$\begin{aligned} h(T'[r(n)]) &= h(T'[|f(T[n])|]) \\ &= h(f(T[n])) && \text{as } T' = f(T[n_0]) \diamond T \\ &= h'(T[n]). \end{aligned}$$

Case 2: $n > n_0$. Then we get

$$\begin{aligned} h(T'[r(n)]) &= h(T'[r(n_0) + n - n_0]) \\ &= h(T'[|f(T[n_0])| + n - n_0]) \\ &= h(f(T[n_0]) \diamond T[n - n_0]) && \text{as } T' = f(T[n_0]) \diamond T \\ &= h(f(T[n_0])) && \text{as } f(T[n_0]) \text{ is a locking sequence of } h \\ &= h'(T[n]). \end{aligned}$$

Thus, all that remains to be shown is that $r \in \vec{R}$. Obviously, r is non-decreasing. Especially, we have that r is strongly monotone increasing for all $n > n_0$. Thus we have, for all m, $\forall^\infty n : r(n) \ge m$. Finally we show that $\text{content}(T'[r(n)]) \subseteq \text{content}(T[n])$. From the construction of f we have, for all $n \le n_0$, $\text{content}(T'[|f(T[n])|]) \subseteq \text{content}(T[n])$. From the construction of r and T' we get that, for all n, $\text{content}(T'[r(n)]) \subseteq \text{content}(T[n])$. □

Next we define semantic and pseudo-semantic restrictions introduced in [14]. Intuitively, semantic restrictions allow one to replace hypotheses by equivalent ones; pseudo-semantic restrictions allow the same, as long as no new mind changes are introduced.

Definition 3.5. For all total functions $p \in \mathfrak{P}$, we let

$$\text{Sem}(p) = \{p' \in \mathfrak{P} \mid \forall i : W_{p(i)} = W_{p'(i)}\};$$
$$\text{Mc}(p) = \{p' \in \mathfrak{P} \mid \forall i : p'(i) \neq p'(i+1) \Rightarrow p(i) \neq p(i+1)\}.$$[2]

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

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

We note that the intersection of two (pseudo-)semantic learning restrictions is again (pseudo-)semantic. All learning restrictions considered in this paper are pseudo-semantic, and all except **Conv**, **SNU**, **SDec** and **Ex** are semantic.

The next lemma shows that, for every pseudo-semantic learning restriction, learning can be done syntactically decisively.

Lemma 3.6. *Let δ be a pseudo-semantic learning criterion. Then we have*

$$[\mathcal{R}\mathbf{TxtG}\delta] = [\mathcal{R}\mathbf{TxtGSynDec}\delta].$$

---

[2] Note that "Sem" is mnemonic for "semantic" and "Mc" for "mind change".

Proof. Let a **TxtG**δ-learner $h \in \mathcal{R}$ be given. We define a learner $h' \in \mathcal{R}$ such that, for all σ,

$$h'(\sigma) = \begin{cases} \text{pad}(h(\sigma), \sigma), & \text{if } \sigma = \lambda \text{ or } h(\sigma) \neq h(\sigma^-); \\ h'(\sigma^-), & \text{otherwise.} \end{cases}$$

The correctness of this construction is straightforward to check. □

As **SynDec** is a delayable learning criterion, we get the following corollary by taking Theorems 3.2 and 3.4 and Lemma 3.6 together. We will always assume our learners to be in this normal form in this paper.

Corollary 3.7. *Let δ be pseudo-semantic and delayable. Then* **TxtGδEx** *allows for strongly locking learning by a syntactically decisive total learner.*

Fulk showed that any **TxtGEx**-learner can be (effectively) turned into an equivalent learner with many useful properties, including strongly locking learning [8]. One of the properties is called *order-independence*, meaning that on any two texts for a target language the learner converges to the same hypothesis. Another property is called *rearrangement-independence*, where a learner h is rearrangement-independent if there is a function f such that, for all sequences σ, $h(\sigma) = f(\text{content}(\sigma), |\sigma|)$ (intuitively, rearrangement independence is equivalent to the existence of a partially set-driven learner for the same language). We define the collection of all the properties which Fulk showed a learner can have to be the *Fulk normal form* as follows.

Definition 3.8. We say a **TxtGEx**-learner h is in *Fulk normal form* if (1)–(5) hold.

1. h is order-independent.
2. h is rearrangement-independent.
3. If h **TxtGEx**-learns a language L from some text, then h **TxtGEx**-learns L.
4. If there is a locking sequence of h for some L, then h **TxtGEx**-learns L.
5. For all $L \in$ **TxtGEx**(h), h is strongly locking on L.

Fulk showed the following Theorem.

Theorem 3.9 *([8, Theorem 13]). Every* **TxtGEx***-learnable set of languages has a* **TxtGEx***-learner in Fulk normal form; furthermore, any given* **TxtGEx***-learner can be* constructively *turned into an equivalent* **TxtGEx***-learner in Fulk normal form.*

4. Full-information learning

In this section we consider various versions of cautious learning and show that all of our variants are either no restriction to learning, or equivalent to conservative learning as is shown in Fig. 5.

Additionally, we will show that every cautious **TxtGEx**-learnable language is conservative **TxtGEx**-learnable which implies that [**TxtGConvEx**], [**TxtGWMonEx**] and [**TxtGCautEx**] are equal. Last, we will separate these three learning criteria from strongly decisive **TxtGEx**-learning and show that [**TxtGSDecEx**] is a proper superset.

Theorem 4.1. *We have that any conservative learner can be assumed cautious and strongly decisive, i.e.*

$$[\textbf{TxtGConvEx}] = [\textbf{TxtGConvSDecCautEx}].$$

Proof. Let $h \in \mathcal{R}$ and $\mathcal{L}$ be such that h **TxtGConvEx**-learns $\mathcal{L}$. We define, for all σ, a set $M(\sigma)$ as follows

$$M(\sigma) = \{\tau \mid \tau \subseteq \sigma \ \wedge \ \forall x \in \text{content}(\tau) : \Phi_{h(\tau)}(x) \leq |\sigma|\}.$$

We let, for all σ,

$$h'(\sigma) = h(\max(M(\sigma))).$$

Let T be a text for a language $L \in \mathcal{L}$. We first show that h' **TxtGEx**-learns L from the text T. As h **TxtGConvEx**-learns L, there are n and e such that, for all $n' \geq n$, $h(T[n]) = h(T[n']) = e$ and $W_e = L$. Thus, there is $m \geq n$ such that, for all $x \in \text{content}(T[n])$, $\Phi_{h(T[n])}(x) \leq m$ and therefore, for all $m' \geq m$, $h'(T[m]) = h'(T[m']) = e$.

Next we show that h' is strongly decisive and conservative; for that we show that, with every mind change, there is a new element of the target included in the conjecture which is currently not included but is included in all future conjectures; it is easy to see that this property implies both caution and strong decisiveness. Let i and i' be such that $\max(M(T[i'])) = T[i]$. This implies that

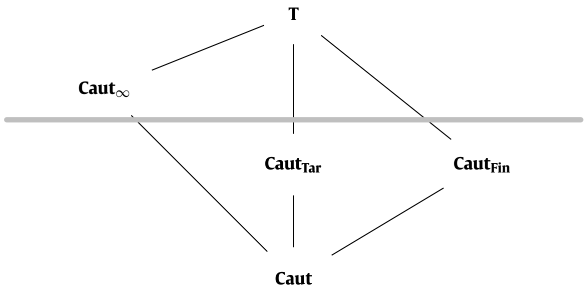


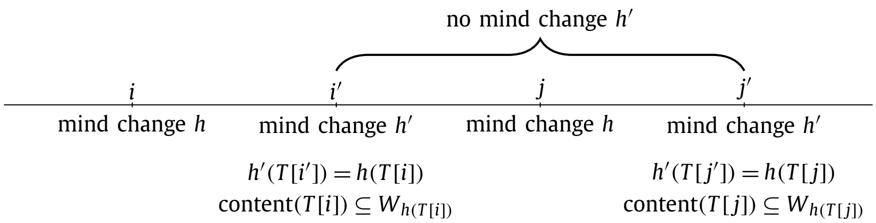
Fig. 5. Relation of different variants of cautious learning. A black line indicates inclusion (bottom to top); all and only the black lines meeting the gray line are proper inclusions.

$$\mathrm{content}(T[i]) \subseteq W_{h'(T[i'])}.$$

Let $j' > i'$ such that $h'(T[i']) \neq h'(T[j'])$. Then there is $j > i$ such that $\max(M(T[j'])) = T[j]$ and therefore

$$\mathrm{content}(T[j]) \subseteq W_{h'(T[j'])}.$$

Note that in the following diagram j could also be between i and i'.



As h is conservative and $\mathrm{content}(T[i]) \subseteq W_{h(T[i])}$, there exists ℓ such that $i < \ell < j$ and $T(\ell) \notin W_{h(T[i])}$. Then we have, for all $n \geq j'$, $T(\ell) \in W_{h'(T[n])}$ as $T(\ell) \in \mathrm{content}(max(M(T[j']))) \subseteq \mathrm{content}(max(M(T[n])))$ and $\mathrm{content}(max(M(T[n]))) \subseteq W_{h'(T[n])}$.

Obviously h' is conservative as it only outputs (delayed) hypotheses of h (and maybe skip some) and h is conservative. $\square$

In the following we consider three new learning restrictions. The learning restriction $\mathbf{Caut_{Fin}}$ means that the learner never returns a hypothesis for a finite set that is a proper subset of a previous hypothesis. $\mathbf{Caut}_\infty$ is the same restriction for infinite hypotheses. With $\mathbf{Caut_{Tar}}$ the learner is not allowed to ever output a hypothesis that is a proper superset of the target language that is learned.

Definition 4.2.

$$\mathbf{Caut_{Fin}}(p, T) \Leftrightarrow [\forall i < j : W_{p(j)} \subset W_{p(i)} \Rightarrow W_{p(j)} \text{ is infinite}]$$
$$\mathbf{Caut}_\infty(p, T) \Leftrightarrow [\forall i < j : W_{p(j)} \subset W_{p(i)} \Rightarrow W_{p(j)} \text{ is finite}]$$
$$\mathbf{Caut_{Tar}}(p, T) \Leftrightarrow [\forall i : \neg(\mathrm{content}(T) \subset W_{p(i)})]$$

The proof of the following theorem is essentially the same as given in [19] to show that cautious learning is a proper restriction of **TxtGEx**-learning, we now extend it to strongly decisive learning. Note that a different extension was given in [3] (with an elegant proof exploiting the undecidability of the halting problem), pertaining to *behaviorally correct* learning. The proof in [3] as well as our proof would also carry over to the combination of these two extensions.

Theorem 4.3. *There is a class of languages that is* **TxtGSDecMonEx**-*learnable, but not* **TxtGCautEx**-*learnable.*

Proof. Let h be a **Psd**-learner as follows,

$$\forall D, t : h(D, t) = \varphi_{\max(D)}(t),$$

and $\mathcal{L} = \mathbf{TxtPsdSDecMonEx}(h)$. Suppose $\mathcal{L}$ is **TxtGCautEx**-learnable through learner $h' \in \mathcal{R}$. We define, for all σ and t, the total computable predicate $Q(\sigma, t)$ as

$$Q(\sigma, t) \Leftrightarrow \mathrm{content}(\sigma) \subset W^t_{h'(\sigma)}.$$

We let ind be such that, for every set D, $W_{\mathrm{ind}(D)} = D$. Using **ORT** we define p and $e \in \mathcal{R}$ strictly monotone increasing such that for all n and t,

$$W_p = \mathrm{range}(e);$$

$$\varphi_{e(n)}(t) = \begin{cases} \mathrm{ind}(\mathrm{content}(e[n+1])), & \text{if } Q(e[n+1], t); \\ p, & \text{otherwise.} \end{cases}$$

Case 1: For all n and t, $Q(e[n+1], t)$ does not hold. Then we have $\varphi_{e(n)}(t) = p$ for all n, t. Thus $W_p \in \mathcal{L}$, as for any $D \subseteq W_p$, $h(D, t) = \varphi_{\max(D)}(t) = p$. But h' does not **TxtGCautEx**-learn W_p from the text e as, for all n and t, content$(e[n])$ is not a proper subset of $W^t_{h'(e[n])}$ although W_p is infinite.

Case 2: There is a minimal n and t such that $Q(e[n+1], t)$ holds. Then we have content$(e[n+1]) \in \mathcal{L}$ as we will show now. Let T be a text for content$(e[n+1])$. As e is monotone increasing we have that $e(n)$ is the maximal element in content$(e[n+1])$. Additionally, for all $t' \geq t$, we have $\varphi_{e(n)}(t') = \varphi_{e(n)}(t) = \mathrm{ind}(\mathrm{content}(e[n+1]))$. As h makes only one mind change the restrictions of strong decisiveness and monotonicity hold. Thus, there is n_0 such that, for all $n \geq n_0$, $h(\mathrm{content}(T[n]), n) = h(\mathrm{content}(T[n_0]), n_0) = \mathrm{ind}(\mathrm{content}(e[n+1]))$, i.e. content$(e[n+1]) \in \mathcal{L}$.

The learner h' does not **TxtGCautEx**-learn content$(e[n+1])$ since we know from the predicate Q that content$(e[n+1]) \subset W_{h'(e[n+1])}$ and the cautious learner h' must not change to a proper subset of a previous hypothesis. $\square$

The following theorem contradicts a theorem given as an exercise in [19] (Exercise 4.5.4B).

Theorem 4.4. *For $\delta \in \{\mathbf{Caut}, \mathbf{Caut_{Tar}}, \mathbf{Caut_{Fin}}\}$ we have*

$$[\mathbf{TxtG}\delta\mathbf{Ex}] = [\mathbf{TxtGConvEx}].$$

Proof. We get the inclusion [**TxtGConvEx**] $\subseteq$ [**TxtGCautEx**] as a direct consequence from Theorem 4.1. Obviously we have [**TxtGCautEx**] $\subseteq$ [$\mathbf{TxtGCaut_{Tar}Ex}$] and [**TxtGCautEx**] $\subseteq$ [$\mathbf{TxtGCaut_{Fin}Ex}$]. Thus, it suffices to show [$\mathbf{TxtG}\delta\mathbf{Ex}$] $\subseteq$ [**TxtGConvEx**].

Let $\mathcal{L}$ be $\mathbf{TxtG}\delta\mathbf{Ex}$-learnable by a syntactically decisive learner $h \in \mathcal{R}$ (see Corollary 3.7). Using the S-m-n Theorem we get a function $p \in \mathcal{R}$ such that, for all σ,

$$W_{p(\sigma)} = \bigcup_{t \in \mathbb{N}} \begin{cases} W^t_{h(\sigma)}, & \text{if } \forall \rho \in (W^t_{h(\sigma)})^*, |\sigma \diamond \rho| \leq t : h(\sigma \diamond \rho) = h(\sigma); \\ \emptyset, & \text{otherwise.} \end{cases}$$

We let $W_{p(\sigma),t'}$ be $W_{p(\sigma)}$ where only the union over $t \leq t'$ is considered.

We let Q be the following computable predicate.

$$Q(\hat{\sigma}, \sigma) \Leftrightarrow h(\hat{\sigma}) \neq h(\sigma) \ \wedge \ \mathrm{content}(\sigma) \not\subseteq W_{p(\hat{\sigma}),|\sigma|-1}.$$

For given sequences σ and τ we say $\tau \preceq \sigma$ if

$$\mathrm{content}(\tau) \subseteq \mathrm{content}(\sigma) \ \wedge \ |\tau| \leq |\sigma|.$$

This means that, for every σ, the set of all τ such that $\tau \preceq \sigma$ is finite and computable. We define a learner h' such that $h'(\sigma)$ is of the form $p(\hat{\sigma})$ for some $\hat{\sigma}$ satisfying content$(\hat{\sigma}) \subseteq$ content(σ). For a given sequence $\sigma \neq \lambda$ let $\hat{\sigma}$ be such that $h'(\sigma^-) = p(\hat{\sigma})$.

$$\forall \sigma : h'(\sigma) = \begin{cases} p(\lambda), & \text{if } \sigma = \lambda; \\ p(\tau \diamond \sigma), & \text{else, if } \exists \tau, \hat{\sigma} \subseteq \tau \preceq \sigma : Q(\hat{\sigma}, \tau);^{3} \\ h'(\sigma^-), & \text{otherwise.} \end{cases}$$

This means h' only changes its hypothesis if Q ensures that h made a mind change and that the previous hypothesis does not contain something of the new input data. We first show that h' is conservative. Let σ and $\hat{\sigma}$ be such that $h'(\sigma^-) = p(\hat{\sigma})$ and let $\tau \preceq \sigma$ be such that $Q(\hat{\sigma}, \tau)$ and $\hat{\sigma} \subseteq \tau$. Suppose now, by way of contradiction, content$(\sigma) \subseteq W_{h'(\sigma^-)} = W_{p(\hat{\sigma})}$. We now have

$$\mathrm{content}(\tau) \subseteq \mathrm{content}(\sigma) \subseteq W_{h'(\sigma^-)} = W_{p(\hat{\sigma})} \subseteq W_{h(\hat{\sigma})}.$$

Let t_0 be minimal such that content$(\tau) \subseteq W^{t_0}_{h(\hat{\sigma})}$. From $Q(\hat{\sigma}, \tau)$ we know

$$\mathrm{content}(\tau) \not\subseteq W_{p(\hat{\sigma}),|\tau|-1};$$

[3] We choose the least such τ, if existent.

this shows $t_0 \geq |\tau|$. Since we chose t_0 minimally and content$(\tau) \subseteq W_{p(\hat{\sigma})}$, we have that, in the definition of $W_{p(\hat{\sigma})}$ the first case holds at least until $t = t_0$, as otherwise we would not include all of content(τ). This shows

$$\forall \rho \in (W^{t_0}_{h(\hat{\sigma})})^*, |\hat{\sigma} \diamond \rho| \leq t_0 : h(\hat{\sigma} \diamond \rho) = h(\hat{\sigma}).$$

This is a contradiction to the fact that we can choose ρ such that $\hat{\sigma} \diamond \rho = \tau$, i.e.

$$\tau \in (W^{t_0}_{h(\hat{\sigma})})^* \wedge |\tau| \leq t_0 \wedge \hat{\sigma} \subseteq \tau \wedge h(\tau) \neq h(\hat{\sigma}).$$

Thus, h' is conservative.

Second, we will show that h' converges on any text T for a language $L \in \mathcal{L}$. Let $L \in \mathcal{L}$ and T be a text for L. Thus, h converges on T. Suppose h' does not converge on T. Let $(p(\sigma_i))_{i \in \mathbb{N}}$ the corresponding sequence of hypotheses. Then $T' = \bigcup_{i \in \mathbb{N}} \sigma_i$ is a text for L as for every $i \in \mathbb{N}$, $T(i)$ is included once a mind change after seeing $T(i)$ occurs. As h' infinitely often changes its mind, we have that, for infinitely many σ_i, there is τ_i such that $\sigma_i \subseteq \tau_i \subseteq \sigma_{i+1}$ with $Q(\sigma_i, \tau_i)$. As $Q(\sigma_i, \tau_i)$ means that $h(\sigma_i) \neq h(\tau_i)$, h diverges on T', a contradiction.

Third we will show that h' converges to a correct hypothesis. Let σ be such that h' converges to $p(\sigma)$ on T. Thus, we have that, for all $\tau \supseteq \sigma$ with content$(\tau) \subseteq L$, we have $\neg Q(\sigma, \tau)$, i.e.

$$h(\sigma) = h(\tau) \ \vee \ \text{content}(\tau) \subseteq W_{p(\sigma), |\tau|-1}.$$

We consider the following two cases.

Case 1: For all $\tau \supseteq \sigma$ with content$(\tau) \subseteq L$, $h(\sigma) = h(\tau)$. Then σ is a locking sequence for h on L. In particular $W_{h(\sigma)} = L$ and thus $W_{p(\sigma)} = W_{h(\sigma)} = L$.

Case 2: There is a $\tau \supseteq \sigma$ with content$(\tau) \subseteq L$ and $h(\sigma) \neq h(\tau)$. Let any such τ of *minimal length* be fixed. For all $x \in L$ we have, as h is syntactically decisive, $h(\sigma) \neq h(\tau \diamond x)$; as $\neg Q(\sigma, \tau \diamond x)$ we get content$(\tau \diamond x) \subseteq W_{p(\sigma), |\tau|} \subseteq W_{h(\sigma)}$. This shows $L \subseteq W_{p(\sigma)} \subseteq W_{h(\sigma)}$ and also L finite, as $W^{|\tau|}_{h(\sigma)}$ is finite.

(a) $\delta = \mathbf{Caut}$. We have that the learner must not change to a proper subset of a previous hypothesis and this means that $W_{h(\sigma)} = L$.

(b) $\delta = \mathbf{Caut_{Tar}}$. The learner h never returns a hypothesis which is a proper superset of the language that is learned. Thus $W_{h(\sigma)} = L$.

(c) $\delta = \mathbf{Caut_{Fin}}$. As h must not change to a finite subset of a previous hypothesis and L is finite, we get $W_{h(\sigma)} = L$.

In either case we now have

$$L = W_{p(\sigma), |\tau|} = W_{h(\sigma)}.$$

Thus, $W_{p(\sigma)} = L$. □

From the definitions of the learning criteria we have [**TxtGConvEx**] ⊆ [**TxtGWMonEx**]. Using Theorem 4.4 and the equivalence of weakly monotone and conservative learning (using **G**) [16,12], we get the following.

Corollary 4.5. *We have*

[**TxtGConvEx**] = [**TxtGWMonEx**] = [**TxtGCautEx**].

Using Corollary 4.5 and Theorem 4.1 we get that weakly monotone **TxtGEx**-learning is included in strongly decisive **TxtGEx**-learning. Theorem 4.3 shows that this inclusion is proper.

Corollary 4.6. *We have*

[**TxtGWMonEx**] ⊂ [**TxtGSDecEx**].

The next theorem is the last theorem of this section and shows that forbidding to go down to strict *infinite* subsets of previously conjectured sets is no restriction.

Theorem 4.7. *We have*

[$\mathbf{TxtGCaut_\infty Ex}$] = [**TxtGEx**].

Proof. Obviously we have [$\mathbf{TxtGCaut_\infty Ex}$] ⊆ [**TxtGEx**]. Thus, we have to show that [**TxtGEx**] ⊆ [$\mathbf{TxtGCaut_\infty Ex}$]. Let $\mathcal{L}$ be a set of languages and h be a learner such that h **TxtGEx**-learns $\mathcal{L}$ and h is strongly locking on $\mathcal{L}$ (see Corollary 3.7). We define, for all σ and t, the set M^t_σ such that

$$M^t_\sigma = \{\tau \mid \tau \in \mathbb{S}\text{eq}(W^t_{h(\sigma)} \cup \text{content}(\sigma)) \ \wedge \ |\sigma \diamond \tau| \leq t\}.$$

Using the S-m-n Theorem we get a function $p \in \mathcal{R}$ such that

$$\forall \sigma : W_{p(\sigma)} = \mathrm{content}(\sigma) \bigcup_{t \in \mathbb{N}} \begin{cases} W^t_{h(\sigma)}, & \text{if } \forall \rho \in M^t_\sigma : h(\sigma \diamond \rho) = h(\sigma); \\ \emptyset, & \text{otherwise.} \end{cases}$$

We define a learner h' as

$$\forall \sigma : h'(\sigma) = \begin{cases} p(\sigma), & \text{if } h(\sigma) \neq h(\sigma^-); \\ h'(\sigma^-), & \text{otherwise.} \end{cases}$$

We will show now that the learner h' $\mathbf{TxtGCaut}_\infty\mathbf{Ex}$-learns $\mathcal{L}$. Let $L \in \mathcal{L}$ and a text T for L be given. As h is strongly locking there is n_0 such that, for all $\tau \in \mathbb{S}\mathrm{eq}(L)$, $h(T[n_0] \diamond \tau) = h(T[n_0])$ and $W_{h(T[n_0])} = L$. Thus we have, for all $n \geq n_0$, $h'(T[n]) = h'(T[n_0])$ and $W_{h'(T[n_0])} = W_{p(T[n_0])} = W_{h(T[n_0])} = L$. To show that the learning restriction $\mathbf{Caut}_\infty$ holds, we assume that there are $i < j$ such that $W_{h'(T[j])} \subset W_{h'(T[i])}$ and $W_{h'(T[j])}$ is infinite. W.l.o.g. j is the first time that h' returns the hypothesis $W_{h'(T[j])}$. Let τ be such that $T[i] \diamond \tau = T[j]$. From the definition of the function p we get that $\mathrm{content}(T[j]) \subseteq W_{h'(T[j])} \subseteq W_{h'(T[i])}$. Thus, $\mathrm{content}(\tau) \subseteq W_{h'(T[i])} = W_{p(T[i])}$ and therefore $W_{p(T[i])}$ is finite, a contradiction to the assumption that $W_{h'(T[j])}$ is infinite. □

5. Decisiveness

In this section the goal is to show that decisive and strongly decisive learning separate (see Theorem 5.3). For this proof we adapt a technique known in computability theory as a "priority argument" (note, though, that we are not dealing with oracle computations). In order to illustrate the proof with a simpler version, we first reprove that decisiveness is a restriction to **TxtGEx**-learning (as shown in [3]).

For both proofs we need the following lemma, a variant of which is given in [3] for the case of decisive learning; it is easy to see that the proof from [3] also works for the cases we consider here.

Lemma 5.1 *([3]). Let $\mathcal{L}$ be such that $\mathbb{N} \notin \mathcal{L}$ and, for each finite set D, there are only finitely many $L \in \mathcal{L}$ with $D \nsubseteq L$. Let $\delta \in \{\mathbf{Dec}, \mathbf{SDec}\}$. Then, if $\mathcal{L}$ is $\mathbf{TxtG}\delta\mathbf{Ex}$-learnable, it is so learnable by a learner which never outputs an index for $\mathbb{N}$.*

Now we get to the theorem regarding decisiveness. Its proof is an adaptation of the proof given in [3] (in fact it uses the exact same idea), rephrased as a priority argument. This rephrased version will be modified later to prove the separation of decisive and strongly decisive learning.

Theorem 5.2 *([3]). We have*

$$[\mathbf{TxtGDecEx}] \subset [\mathbf{TxtGEx}].$$

Proof. For this proof we will employ a technique from computability theory known as *priority argument*. For this technique, one has a set of *requirements* (we will have one for each $e \in \mathbb{N}$) and a *priority* on requirements (we will prioritize smaller e over larger). One then tries to fulfill requirements one after the other in an iterative manner (fulfilling the unfulfilled requirement of highest priority without violating requirements of higher priority) so that, in the limit, the entire infinite list of requirements will be fulfilled.

We apply this technique in order to construct a learner $h \in \mathcal{P}$ (and a corresponding set of learned sets $\mathcal{L} = \mathbf{TxtGEx}(h)$). Thus, we will give requirements which will depend on the h to be constructed. In particular, we will use a list of requirement $(R_e)_{e\in\mathbb{N}}$, where lower e have higher priority. For each e, R_e will correspond to the fact that learner φ_e is not a suitable decisive learner for $\mathcal{L}$. We proceed with the formal argument.

For each e, let Requirement R_e be the disjunction of the following three predicates depending on the h to be constructed.

(i) $\exists x$: $\forall \sigma \in \mathbb{S}\mathrm{eq}(\mathbb{N} \setminus \{x\}) : \varphi_e(\sigma)\uparrow \vee W_{\varphi_e(\sigma)} \neq \mathbb{N} \setminus \{x\}$ and h learns $\mathbb{N} \setminus \{x\}$.
(ii) $\exists \sigma \in \mathbb{S}\mathrm{eq} : \mathrm{content}(\sigma) \subset W_{\varphi_e(\sigma)}$ and h learns $W_{\varphi_e(\sigma)}$ and some D with $\mathrm{content}(\sigma) \subseteq D \subset W_{\varphi_e(\sigma)}$.
(iii) $\exists \sigma \in \mathbb{S}\mathrm{eq} : W_{\varphi_e(\sigma)} = \mathbb{N}$.

If all $(R_e)_{e\in\mathbb{N}}$ hold, then every learner which never outputs an index for $\mathbb{N}$ fails to learn $\mathcal{L}$ decisively as follows. For each learner φ_e which never outputs an index for $\mathbb{N}$, either (i) of R_e holds, implying that some co-singleton is learned by h but not by φ_e. Or (ii) holds, then there is a σ on which φ_e generalizes, but will later have to abandon this correct conjecture $p = \varphi_e(\sigma)$ in order to learn some finite set D; as, after the change to a hypothesis for D, the text can still be extended to a text for W_p, the learner is not decisive.[4]

[4] One might wonder why the U-shape can be achieved on a language which is to be learned: after all, those can be avoided, according to the theorem that non-U-shaped learning is not a restriction to **TxtGEx** [3]. However, the price for avoiding it is to output a conjecture for $\mathbb{N}$.

Thus, all that remains is to construct h in a way that all of $(R_e)_{e\in\mathbb{N}}$ are fulfilled. In order to coordinate the different requirements when constructing h on different inputs, we will divide the set of all possible input sequences into infinitely many segments, of which every requirement can "claim" up to two at any point of the algorithm defining h; the chosen segments can change over the course of the construction, and requirements of higher priority might "take away" segments from requirements with lower priority (but not vice versa). We follow [3] with the division of segments: For any set $A \subset \mathbb{N}$ we let $\mathrm{id}(A) = \min(\mathbb{N} \setminus A)$ be the *ID of A*; for ease of notation, for each finite sequence σ, we let $\mathrm{id}(\sigma) = \mathrm{id}(\mathrm{content}(\sigma))$. For each s, the sth segment contains all σ with $\mathrm{id}(\sigma) = s$. We note that id is *monotone*, i.e.

$$\forall A, B \subset \mathbb{N} : A \subseteq B \Rightarrow \mathrm{id}(A) \leq \mathrm{id}(B). \tag{1}$$

The first way of ensuring some requirement R_e is via (i); as this part itself is not decidable, we will check a "bounded" version thereof. We define, for all e, t, s,

$$P_{e,t}(s) \Leftrightarrow (\forall \sigma \in \mathbb{S}\mathrm{eq}_{\leq t} \mid \mathrm{id}(\sigma) = s)\ \Phi_e(\sigma) > t \vee \mathrm{content}(\sigma) \not\subset W^t_{\varphi_e(\sigma)}.$$

For any e, if we can find an s such that, for all t, we have $P_{e,t}(s)$, then it suffices to make h learn $\mathbb{N} \setminus \{s\}$ in order to fulfill R_e via part (i); this requires control over segment s in defining h.

Note that, if we ever cannot take control over some segment because some requirement with higher priority is already in control, then we will try out different s (only finitely many are blocked).

If we ever find a t such that $\neg P_{e,t}(s)$, then we can work on fulfilling R_e via (ii), as we directly get a σ where φ_e over the content generalizes. In order to fulfill R_e via (ii) we have to choose a finite set D with $\mathrm{content}(\sigma) \subseteq D \subset W_{\varphi_e(\sigma)}$. We will then take control over the segments corresponding to $\mathrm{id}(D)$ and $\mathrm{id}(W^t_{\varphi_e(\sigma)})$ (for growing t), *but not necessarily over segment* s, and thus establish R_e via (ii). Note that, again, the segments we desire might be blocked; but only finitely many are blocked, and we require control over $\mathrm{id}(D)$ and $\mathrm{id}(W^t_{\varphi_e(\sigma)})$, both of which are at least s (this follows from id being monotone, see Equation (1), and from $\mathrm{content}(\sigma) \subseteq D \subset W^t_{\varphi_e(\sigma)}$); thus, we can always find an s for which we can either follow our strategy for (i) or for (ii) as just described.

It is tempting to choose simply $D = \mathrm{content}(\sigma)$, this fulfills all desired properties. The main danger now comes from the possibility of $\varphi_e(\sigma)$ being an index for $\mathbb{N}$: this will imply that, for growing t, $y = \mathrm{id}(W^t_{\varphi_e(\sigma)})$ will also be growing indefinitely. Of course, there is no problem with satisfying R_e, it now holds via (iii); but as soon as at least two requirements will take control over segments y for indefinitely growing y, they might start blocking each other (more precisely, the requirement of higher priority will block the one of lower priority). We now need to know something about our later analysis: we will want to make sure that every requirement R_e either (a) converges in which segments to control or (b) for all n, there is a time t in the definition of h after which R_e will never have control over any segment corresponding to IDs $\leq n$; in fact, we will show this later by induction (see Claim 2). Any requirement which takes control over segments y for indefinitely growing y might be blocked infinitely often, and thus forced to try out different s for fulfilling R_e, including returning to s that were abandoned previously because of (back then) being blocked by a requirement of higher priority. Thus, such a requirement would fulfill neither (a) nor (b) from above. We will avoid this problem by *not* choosing $D = \mathrm{content}(\sigma)$, but instead choosing a D which grows in ID along with the corresponding $W^t_{\varphi_e(\sigma)}$. The idea is to start with $D = \mathrm{content}(\sigma)$ and then, as $W^t_{\varphi_e(\sigma)}$ grows, add more elements. For this we make some definitions as follows.

For a finite sequence σ we let $\mathrm{id}'(\sigma)$ be the least element not in $\mathrm{content}(\sigma)$ which is larger than all elements of $\mathrm{content}(\sigma)$. For any finite sequence σ and $e, t \geq 0$ we let $D^t_{e,\sigma}$ be such that

$$D^t_{e,\sigma} = \begin{cases} \mathrm{content}(\sigma), & \text{if } \mathrm{id}(W^t_{\varphi_e(\sigma)}) \leq \mathrm{id}'(\sigma); \\ \{0, \ldots, \mathrm{id}(W^t_{\varphi_e(\sigma)}) - 2\}, & \text{otherwise.} \end{cases}$$

For all e, t and σ with $\mathrm{content}(\sigma) \subset W_{\varphi_e(\sigma)}$ we have

$$\mathrm{content}(\sigma) \subseteq D^t_{e,\sigma} \subset W_{\varphi_e(\sigma)}. \tag{2}$$

Thus, we will use the sets $D^t_{e,\sigma}$ to satisfy (ii) of R_e (in place of D).

We now have all parts that are required to start giving the construction for h. In that construction we will make use of a subroutine which takes as inputs a set B of blocked indices, a requirement e and a time bound t, and which finds triples (x, y, σ) with $x, y \notin B$ such that

$$P_{e,t}(x) \text{ or } \left[\mathrm{content}(\sigma) \subset W^t_{\varphi_e(\sigma)} \wedge \mathrm{id}(D^t_{e,\sigma}) = x \wedge \mathrm{id}(W^t_{\varphi_e(\sigma)}) = y\right]. \tag{3}$$

We call (x, y, σ) fulfilling Equation (3) for given t and e a *t-witness for* R_e. The subroutine is called `findWitness` and is given in Algorithm 1.

We now formally show termination and correctness of our subroutine.

Claim 1. *Let e, t and a finite set B be given. The algorithm* `findWitness` *on (B, e, t) terminates and returns a t-witness (x, y, σ) for R_e such that $x, y \notin B$.*

Algorithm 1: `findWitness`(B, e, t).

```
1  Input: finite set B, e, t ∈ ℕ;
2  for s = 0 to max(B) + 1 do
3      if P_{e,t}(s) and s ∉ B then
4          return (s, s, 0);
5      else if ¬P_{e,t}(s) then
6          Let σ be minimal with id(σ) = s and content(σ) ⊂ W^t_{φ_e(σ)};
7          x ← id(D^t_{e,σ});
8          y ← id(W^t_{φ_e(σ)});
9          if x ∉ B and y ∉ B then
10             return (x, y, σ);
11 return error;
```

Algorithm 2: Priority construction **Dec**.

```
1  for t = 0 to ∞ do
2      for e = 0 to t do
3          if t = 0, w_e(t − 1) is undefined or w_e(t − 1) is not (e, t)-legal then
4              Let B be the set of IDs blocked by any e' < e;
5              (x, y, σ) ← findWitness(B, e, t);
6          else
7              (x, y, σ) ← w_e(t − 1);
8          w_e(t) ← (x, y, σ);
9          if P_{e,t}(x) then
10             foreach τ ∈ Seq_{≤t} with id(τ) = x do
11                 h(τ) ← q(x);
12         else
13             foreach τ ∈ Seq_{≤t} with content(τ) = D^t_{e,σ} do
14                 h(τ) ← p(e, t, σ);
15             foreach τ ∈ Seq_{≤t} with id(τ) = y do
16                 h(τ) ← φ_e(σ);
```

Proof of Claim 1. From the condition in line 5 we see that the search in line 6 is necessarily successful, showing termination. Using the monotonicity of id from Equation (1) on Equation (2) we have that the subroutine `findWitness` cannot return `error` on any arguments (B, e, t): for $s = \max(B) + 1$, we either have $P_{e,t}(s)$ or the x and y chosen are larger than $\mathrm{id}(\sigma) = s > \max(B)$. □ (FOR CLAIM 1)

With the subroutine given above, we now turn to the priority construction for defining h detailed in Algorithm 2. This algorithm assigns witness tuples to more and more requirements, trying to make sure that they are t-witnesses, for larger and larger t. For each e, $w_e(t)$ will be the witness tuple associated with R_e after t iterations (defined for all $t \geq e$). We say that a requirement R_e *blocks* an ID n iff $n \in \{x, y\}$ for the witness tuple $w_e(t) = (x, y, \sigma)$ currently associated with R_e. We say that a tuple (x, y, σ) is *(e, t)-legal* iff it is a t-witness for R_e and x and y are not blocked by any $R_{e'}$ with $e' < e$. Clearly, it is decidable whether a triple is (e, t)-legal.

In order to define the learner h we will need some functions giving us indices for the languages to be learned. To that end, let $p, q \in \mathcal{R}$ (using the S-m-n Theorem) be such that

$$\forall n : W_{q(n)} = \mathbb{N} \setminus \{n\};$$
$$\forall e, t, \sigma : W_{p(e,t,\sigma)} = D^t_{e,\sigma}.$$

Since $D^t_{e,\sigma}$ is computable in t, e, σ, we can choose p such that, for any $t, t', e, e', \sigma, \sigma'$ with $D^t_{e,\sigma} = D^{t'}_{e',\sigma'}$ we have $p(e, t, \sigma) = p(e', t', \sigma')$. To increase readability, we allow assignments to values of h for arguments on which h was already defined previously; in this case, the new assignment has no effect.

Regarding Algorithm 2, note that lines 3–8 make sure that we have an appropriate witness tuple. We will later show that the sequence of assigned witness tuples will converge (for learners never giving a conjecture for $\mathbb{N}$). Lines 9–11 will try to establish the requirement R_e via (i), once this fails it will be established in lines 12–16 via (ii).

After this construction of h, we let $\mathcal{L} = \mathbf{TxtGEx}(h)$ be the target to be learned. First note that the IDs blocked by different requirements are always disjoint (at the end of an iteration of t). As the major part of the analysis, we show the following claim by induction, showing that, for each e, either the triple associated with R_e converges or it grows arbitrarily in both its x and y value (this is what we earlier had to carefully choose the D for).

Claim 2. *For all e we have R_e and, for all n, there is t_0 such that either*

$$\forall t \geq t_0 : R_e \text{ does not block any ID } \leq n \text{ at iteration } t$$

or

$$\forall t \geq t_0 : w_e(t) = w_e(t_0).$$

Proof of Claim 2. As our induction hypothesis, let e be given such that the claim holds for all $e' < e$.

Case 1: There is t_0 such that $\forall t \geq t_0 : w_e(t) = w_e(t_0)$.

Then, for all t, $(x, y, \sigma) = w_e(t_0)$ is a t-witness for R_e; in the case of $\forall t : P_{e,t}(x)$, we have that, for all but finitely many τ with $\mathrm{id}(\tau) = x$, $h(\tau) = q(x)$, and index for $\mathbb{N} \setminus \{x\}$; this implies $\mathbb{N} \setminus \{x\} \in \mathcal{L}$, which shows R_e.

Otherwise we have, for all $t \geq t_0$, $D^t_{e,\sigma} = D^{t_0}_{e,\sigma}$. Furthermore we get, for all but finitely many τ with $\mathrm{content}(\tau) = D^{t_0}_{e,\sigma}$, $h(\tau) = p(e, t, \sigma)$, and index for $D^{t_0}_{e,\sigma}$; this implies $D^{t_0}_{e,\sigma} \in \mathcal{L}$. Consider now all those τ with $\mathrm{id}(\tau) = y$. If $\mathrm{id}(D^{t_0}_{e,\sigma}) = y$, then h is already defined on infinitely many such τ, namely in case of $\mathrm{content}(\tau) = D^{t_0}_{e,\sigma}$. However, we have that $D^{t_0}_{e,\sigma}$ is a *proper* subset of $W_{\varphi_e(\sigma)}$, which shows that, on any text for $W_{\varphi_e(\sigma)}$, h will eventually only output $\varphi_e(\sigma)$, which gives $W_{\varphi_e(\sigma)} \in \mathcal{L}$ as desired and, thus, R_e.

Case 2: Otherwise.

For each ID s there exists at most finitely many σ with $\mathrm{id}(\sigma) = s$ and σ is used in the witness triple for R_e; this follows from the choice of σ in the subroutine `findWitness` as a minimum, where, for larger t, all previously considered σ are still considered (so that the chosen minimum might be smaller for larger t, but never go up, which shows convergence). A triple is only abandoned if it is not legal any more; this means it is either blocked or it is not a t-witness triple for some t. Using the induction hypothesis, the first can only happen finitely many times for any given tuple. Thus, the witness tuple changes infinitely many times. Also using the induction hypotheses, there is some time t_0 after which all requirements with higher priority either do not block any elements below n or are converged. From the definition of `findWitness`, we now see that both x and y in the witness tuple found for e grow above n. For this we also use our specific choice of D as growing along with the ID of the associated $W^t_{\varphi_e(\sigma)}$ and we use that any witness tuple with a σ with $\mathrm{id}(\sigma) = s$ has x and y value of at least s, due to the monotonicity of id.

To show R_e (we will show (iii)), let t_1 be the maximum over all t_0 existing for the $e' < e$ for which the limiting value of $w_{e'}(\cdot)$ converges, by the induction hypothesis and e. Let $(x, y, \sigma) = w_e(t_1)$ be the t_1-witness triple chosen for R_e in iteration t_1. Suppose, by way of contradiction, that $\varphi_e(\sigma)$ is not an index for $\mathbb{N}$; let $n = \mathrm{id}(W_{\varphi_e(\sigma)})$. Let t_2 be the maximum over all t_0 found by the induction hypothesis for all $e' < e$ with the chosen n. Since the triple (x, y, σ) is (e, t)-legal for all $t \geq t_2$, we get a contradiction to the unbounded growth of the witness triple.

This shows that $\varphi_e(\sigma)$ is an index for $\mathbb{N}$, and thus we have R_e. □ (FOR CLAIM 2)

With the last claim we now see that all requirement are satisfied. This implies that $\mathcal{L}$ cannot be **TxtGDecEx**-learned by a learner never using an index for $\mathbb{N}$ as conjecture.

We have that $\mathbb{N} \notin \mathcal{L}$. Furthermore, for any ID s, there are only finitely many sets in $\mathcal{L}$ with that ID; this implies that, for every finite set D, there are only finitely many elements $L \in \mathcal{L}$ with $D \not\subseteq L$. Thus, using Lemma 5.1, $\mathcal{L}$ is not decisively learnable at all. □

While the previous theorem showed that decisiveness poses a restriction on **TxtGEx**-learning, the next theorem shows that the requirement of strong decisiveness is even more restrictive. The proof follows the proof of Theorem 5.2, with some modifications.

Theorem 5.3. *We have*

$$[\mathbf{TxtGSDecEx}] \subset [\mathbf{TxtGDecEx}].$$

Proof. We use the same language and definitions as in the proof of Theorem 5.2. The idea of this proof is as follows. We build a set $\mathcal{L}$ with a priority construction just as in the proof of Theorem 5.2, the only essential change being in the definition of the hypothesis $p(e, t, \sigma)$: the change from $\varphi_e(\sigma)$ to $p(e, t, \sigma)$ and back to $\varphi_e(\sigma)$ on texts for $W_{\varphi_e(\sigma)}$ is what made $\mathcal{L}$ not decisively learnable. Thus, we will change $p(e, t, \sigma)$ to be a hypothesis for $W_{\varphi_e(\sigma)}$ as well – *as soon as φ_e changed its hypothesis on an extension of σ*, and otherwise it is a hypothesis for $D^t_{e,\sigma}$ as before. This will make h decisive on texts for $W_{\varphi_e(\sigma)}$, but φ_e will not be strongly decisive.

Furthermore, we will make sure that for sequences with ID s, only conjectures for sets with ID s are used, so that indecisiveness can only possibly happen within a segment. Now the last source of $\mathcal{L}$ not being decisively learnable is as follows. When different requirements take turns with being in control over the segment, they might introduce returns to abandoned conjectures. To counteract this, we make sure that any conjecture which is ever abandoned on a segment of ID s is for $\mathbb{N} \setminus \{s\}$, which will give decisiveness.

Algorithm 3: Priority construction **SDec**.

1 **for** $t = 0$ **to** ∞ **do**
2   **for** $e = 0$ **to** t **do**
3   **if** $t = 0$, $w_e(t-1)$ *is undefined or* $w_e(t-1)$ *is not* (e,t)*-legal* **then**
4   Let B be the set of IDs blocked by any $e' < e$;
5       $(x, y, \sigma) \leftarrow \texttt{findWitness}(B, e, t)$;
6   **else**
7       $(x, y, \sigma) \leftarrow w_e(t-1)$;
8     $w_e(t) \leftarrow (x, y, \sigma)$;
9   **if** $P_{e,t}(x)$ **then**
10   **foreach** $\tau \in \mathbb{S}\mathrm{eq}_{\le t}$ *with* $\mathrm{id}(\tau) = x$ **do**
11         $h(\tau) \leftarrow q(x)$;
12   **else**
13   **if** $\exists \tau \in \mathbb{S}\mathrm{eq}_{\le t}(D^t_{e,\sigma}) : \varphi_e(\sigma \diamond \tau){\downarrow_t} \neq \varphi_e(\sigma)$ **then**
14   **foreach** $\tau \in \mathbb{S}\mathrm{eq}_{\le t}$ *with* $\mathrm{id}(\tau) = y$ **do**
15           $h(\tau) \leftarrow g(e, \sigma, y)$;
16   **else**
17   **foreach** $\tau \in \mathbb{S}\mathrm{eq}_{\le t}$ *with* $\mathrm{content}(\tau) = D^t_{e,\sigma}$ **do**
18           $h(\tau) \leftarrow p'(e, t, \sigma)$;

We first define an alternative p' for the function p from that proof with the S-m-n Theorem such that, for all e, t, σ,

$$W_{p'(e,t,\sigma)} = \begin{cases} W_{\varphi_e(\sigma)}, & \text{if } \exists \tau \text{ with content}(\tau) \subseteq D^t_{e,\sigma} : \varphi_e(\sigma \diamond \tau){\downarrow} \neq \varphi_e(\sigma); \\ D^t_{e,\sigma}, & \text{otherwise.} \end{cases}$$

As we have $D^t_{e,\sigma} \subseteq W_{\varphi_e(\sigma)}$, this is a valid application of the S-m-n Theorem. Just as with p in the proof of the previous theorem, since $D^t_{e,\sigma}$ is computable in t, e, σ, we can choose p' such that, for any $t, t', e, e', \sigma, \sigma'$ with $D^t_{e,\sigma} = D^{t'}_{e',\sigma'}$ we have $p'(e, t, \sigma) = p'(e', t', \sigma')$. We also want to replace the output of h according to line 16 of Algorithm 2. To that end, let $g \in \mathcal{R}$ be as given by the S-m-n Theorem such that, for all e and σ,

$$W_{g(e,\sigma,y)} = W_{\varphi_e(\sigma)} \setminus \{y\}.$$

We construct now a learner h again according to a priority construction, as given in Algorithm 3. Note that lines 1–12 are identical with the construction from Algorithm 2 and lines 3–8 again make sure that we have an appropriate witness tuple and lines 9–11 try to establish the requirement R_e via (i). The main difference lies in the way that R_e is established once this fails in lines 12–18 via (ii): Here we need to check for a mind change and adjust what language h should learn accordingly.

It is easy to check that h, on any sequence σ, gives conjectures for languages of the same ID as that of σ. Thus, indecisiveness of h can only occur within a segment.

Next we will modify h to avoid indecisiveness from different requirements taking turns controlling the same segment. With the S-m-n Theorem we let $f \in \mathcal{R}$ be such that, for all σ,

$$W_{f(\sigma)} = \begin{cases} \mathbb{N} \setminus \{\mathrm{id}(\sigma)\}, & \text{if } \exists \tau \text{ with id}(\sigma) \notin \mathrm{content}(\tau) : h(\sigma) \neq h(\sigma \diamond \tau); \\ W_{h(\sigma)}, & \text{otherwise.} \end{cases}$$

Let h' be such that, for all σ,

$$h'(\sigma) = \begin{cases} h'(\sigma^-), & \text{if } \sigma \neq \lambda \text{ and } h(\sigma) = h(\sigma^-); \\ f(\sigma), & \text{otherwise.} \end{cases}$$

We now let $\mathcal{L} = \mathbf{TxtGDecEx}(h')$. It is easy to see that h' is decisive on all texts where it always makes an output, since indecisiveness can again only happen within a segment, and f *poisons* any possible non-final conjectures within a segment.

Let a strongly decisive learner $\bar{h}$ for $\mathcal{L}$ be given which never makes a conjecture for $\mathbb{N}$ (we are reasoning with Lemma 5.1 again). Let e be such that $\varphi_e = \bar{h}$. Reasoning as in the proof of Theorem 5.2, we see that there is a triple (x, y, σ) such that w_e converges to that triple in the construction of h'. If, for all t, $P_{e,t}(x)$, then we have that $\mathbb{N} \setminus \{x\} \in \mathcal{L}$ (on any sequences with ID x, h' gives an output for $\mathbb{N} \setminus \{x\}$, and it converges). Assume now that there is t_0 such that, for all $t \ge t_0$, we have $\neg P_{e,t}(x)$.

Case 1: There is τ with $\mathrm{content}(\tau) \subseteq D^t_{e,\sigma}$ such that $\varphi_e(\sigma \diamond \tau) \neq \varphi_e(\sigma)$.

Let T be a text for $L = W_{\varphi_e(\sigma)}$. Then h' on T converges to an index for L, giving $L \in \mathcal{L}$. But this shows that $\bar{h} = \varphi_e$ was not strongly decisive on any text for L starting with $\sigma \diamond \tau$, a contradiction.

Case 2: Otherwise.

Let T be a text for $L = D^t_{e,\sigma}$. Then h' on T converges to an index for L, giving $L \in \mathcal{L}$. But $\overline{h} = \varphi_e$ converges on any text for L starting with σ to $\varphi_e(\sigma)$, a contradiction to $D^t_{e,\sigma} \subset W_{\varphi_e(\sigma)}$ (so the convergence is not to a correct hypothesis).

In both cases we get the desired contradiction. □

6. Set-driven learning

In this section we give theorems regarding set-driven learning. For this we build on the result that set-driven learning can always be done conservatively [16].

First we show that any conservative set-driven learner can be assumed to be cautious and syntactically decisive, an important technical lemma.

Lemma 6.1. *We have*

$$[\mathbf{TxtSdEx}] = [\mathbf{TxtSdConvSynDecEx}].$$

In other words, every set-driven learner can be assumed conservative and syntactically decisive. Furthermore, we can assume the learner to be syntactically decisive on all *texts, not just texts for learned languages.*

Proof. Let a set-driven learner h be given. Following [16] we can h assume to be conservative. We define a learner h' such that, for all finite sets C,

$$h'(C) = \begin{cases} \mathrm{pad}(h(C), 0), & \text{if } \forall D \subseteq C : h(D) = h(C) \rightarrow \forall D', D \subseteq D' \subseteq C : h(D') = h(D); \\ \mathrm{pad}(h(C), |C| + 1), & \text{otherwise.} \end{cases}$$

Let $\mathcal{L} = \mathbf{TxtSdConvEx}(h)$. We will show that h' is syntactically decisive and $\mathbf{TxtSdConvEx}$-learns $\mathcal{L}$. Let $L \in \mathcal{L}$ be given and let T be a text for L. First, we show that h' $\mathbf{TxtEx}$-learns L from T. As h is a set-driven learner there is n_0 such that $\forall n \geq n_0 : h(\mathrm{content}(T[n_0])) = h(\mathrm{content}(T[n]))$ and $W_{h(\mathrm{content}(T[n_0]))} = L$. We will show that, for all $T[n]$ with $n \geq n_0$, the first condition in the definition of h' holds. Let $n \geq n_0$ and suppose there are D and D' with

$$D \subseteq \mathrm{content}(T[n]),$$
$$h(D) = h(\mathrm{content}(T[n])) = h(\mathrm{content}(T[n_0]))$$

and

$$D \subseteq D' \subseteq \mathrm{content}(T[n]),$$
$$h(D) \neq h(D').$$

As $W_{h(D)} = L$ and h is conservative, h must not change its hypothesis. Thus, for all D' with $D \subseteq D' \subseteq L$ we get $h(D') = h(D)$, a contradiction.

Thus we have, for all $n \geq n_0$,

$$\begin{aligned} h'(\mathrm{content}(T[n])) &= h'(\mathrm{content}(T[n_0])) \\ &= \mathrm{pad}(h(\mathrm{content}(T[n_0])), 0) \end{aligned}$$

and $W_{h'(\mathrm{content}(T[n_0]))} = W_{\mathrm{pad}(h(\mathrm{content}(T[n_0])),0)} = L$, i.e. h' $\mathbf{TxtGEx}$-learns L.

Second, we will show that h' is conservative. Whenever h makes a mind change, h' will also make a mind change; as, for all n, $W_{h(\mathrm{content}(T[n]))} = W_{h'(\mathrm{content}(T[n]))}$, we have that h' is conservative in these cases. Thus, we have to show that h' is conservative whenever it changes its mind because the first condition in the definition does not hold. Let n such that

$$h'(\mathrm{content}(T[n])) \neq h'(\mathrm{content}(T[n-1]))$$

because the first condition in the definition of h' is violated. Let $C = \mathrm{content}(T[n])$. Thus, there are D and D' with $D \subseteq D' \subseteq C$ such that $h(D) = h(C)$ and $h(D') \neq h(C)$. We consider the case that $h(T[n]) = h(T[n-1])$ as otherwise h' is obviously conservative. As h is conservative we can conclude that there is $x \in D'$ such that $x \notin W_{h(D)}$. Otherwise we could construct a text T' for L starting with elements of D on which h would not be conservative. Thus there is $x \in D' \subseteq C$ such that

$$x \notin W_{h(C)} = W_{h(T[n])} = W_{h(T[n-1])} = W_{h'(T[n-1])}$$

and therefore h' is still conservative if it changes its mind.

To show that h' is syntactically decisive let $C \subseteq D \subseteq E$ be such that $h'(C) \neq h'(D)$ and $h'(C) = h'(E)$. We then know that $h'(E)$ is defined by the second branch in the definition of h' as either $h(C) = h(E) \neq h(D)$ or $h(C) = h(D) = h(E)$. As there exist $D' \subseteq D'' \subseteq D$ such that $h(D') = h(D) \neq h(D'')$ we have in both cases E', E'' witnessing that $E' \subseteq E'' \subseteq E$ and $h(E') = h(E) \neq h(E'')$. We therefore get that $C \subset E$. Thus $0 \neq |C| + 1 \neq |E| + 1$ and therefore the second component in pad is different for C and E. This implies that $h'(C) \neq h'(E)$ as pad is 1–1. □

The following Theorem is the main result of this section, showing that set-driven learning can be done not just conservatively, but also strongly decisively and cautiously *at the same time*.

Theorem 6.2. *We have*

$$[\mathbf{TxtSdEx}] = [\mathbf{TxtSdConvSDecCautEx}].$$

Proof. Following [16] we can assume a set-driven learner to be conservative. Let h and $\mathcal{L}$ be such that h **TxtSdConvEx**-learns $\mathcal{L}$ and suppose that h is syntactically decisive on all texts using Lemma 6.1. We define a function p using the S-m-n Theorem such that, for every set D and e,

$$W_{p(D,e)} = D \bigcup_{t\in\mathbb{N}} \begin{cases} \emptyset, & \text{if } \exists D' \subseteq D \cup W_e^t : D' < D \wedge h(D') = e; \\ W_e^t, & \text{if } h(D \cup W_e^t) = e; \\ \emptyset, & \text{otherwise.} \end{cases}$$

We define a function N such that, for any finite set D,

$$N(D) = \{D' \subseteq D \mid h(D) = h(D')\}.$$

We define h', for all finite sets D, as

$$h'(D) = p(\min(N(D)), h(D)).$$

Let $L \in \mathcal{L}$ be given and let T be a text for L. We first show that h' **TxtSdEx**-learns L from T. As h **TxtSdEx**-learns L we know that h is strongly locking on T (this was shown in [6]). Thus there is n_0 such that $T[n_0]$ is a locking sequence for h on L. Let $D' \subseteq L$ be minimal with $h(D') = h(\mathrm{content}(T[n_0]))$ and n_1 such that $D' \subseteq \mathrm{content}(T[n_1])$. Thus we have, for all $n \geq n_1$, $\min(N(\mathrm{content}(T[n]))) = D'$. From the construction of p and h syntactically decisive we get

$$W_{p(D',h(D'))} = W_{h(D')}.$$

This shows that h' **TxtSdEx**-learns L. We proceed by showing the following claim.

Claim 1. *Let $i \leq j$, $D_0 = \mathrm{content}(T[i])$ and $D_1 = \mathrm{content}(T[j])$ and suppose $h'(D_0) \neq h'(D_1)$. Then $(D_1 \cap W_{h'(D_1)}) \setminus W_{h'(D_0)} \neq \emptyset$.*

Proof of Claim 1. Suppose first that $h(D_0) \neq h(D_1)$. From the construction of h' we get that there is $B_1 \subseteq D_1$ with $h'(B_1) = h'(D_1)$ such that h' is consistent on B_1, i.e. $B_1 \subseteq W_{h'(D_1)}$. Additionally we have $B_0 = \min(N(D_0)) \subseteq D_0$ with $h'(B_0) = h'(D_0)$ and $B_0 \subseteq W_{h'(D_0)}$. Suppose, by way of contradiction, $B_1 \subseteq W_{h'(B_0)}$. Thus, there is a t such that $B_1 \subseteq B_0 \cup W^t_{h(B_0)}$ and $h(B_0 \cup W^t_{h(B_0)}) = h(B_0)$. We have that $B_0 \cup B_1$ is a set in between B_0 and $B_0 \cup W^t_{h(B_0)}$, so from h being syntactically decisive, $h(B_1 \cup B_0) = h(B_0)$. Furthermore, $B_0 \cup B_1$ is a set in between B_1 and D_1, so $h(B_1) = h(B_1 \cup B_0) = h(D_1)$; overall $h(D_0) = h(D_1)$, a contradiction. Thus, $B_1 \subseteq (D_1 \cap W_{h'(D_1)})$ but $B_1 \not\subseteq W_{h'(B_0)} = W_{h'(D_0)}$ as desired.

Suppose now $h(D_0) = h(D_1)$. Thus we have $\min(N(D_0)) \neq \min(N(D_1))$; in particular, as $D_0 \subseteq D_1$, we get $\min(N(D_1)) < \min(N(D_0))$. Let $B_1 = \min(N(D_1))$. From the first case of the construction of p we have that $B_1 \not\subseteq W_{p(\min(N(D_0)),h(D_0))} = W_{h'(D_0)}$. As $B_1 \subseteq D_1$, $h'(D_1) = p(B_1, h(D_1))$ and $B_1 \subseteq W_{p(B_1,h(D_1))}$, we have $D_1 \cap W_{h'(D_1)} \setminus W_{h'(D_0)} \neq \emptyset$ as desired. □ (for Claim 1)

From the claim we know that, for any mind change, there is a datum in the input such that any previous conjecture did not contain this datum, showing h' is conservative. Furthermore, also from the claim we see that any new conjecture contains a datum which was not present in any previous conjecture, showing h' to be cautious and strongly decisive. □

7. Monotone learning

In this section we show the hierarchies regarding monotone and strongly monotone learning, simultaneously for the settings of **G** and **Sd** in Theorems 7.1 and 7.2. With Theorems 7.3 and 7.4 we establish that monotone learnability implies strongly decisive learnability.

Theorem 7.1. *There is a class of languages $\mathcal{L}$ that is* **TxtSdMonWMonEx**-*learnable but not* **TxtGSMonEx**-*learnable, i.e.*

$$[\mathbf{TxtSdMonWMonEx}] \setminus [\mathbf{TxtGSMonEx}] \neq \emptyset.$$

Proof. This is a standard proof which we include for completeness. Let $L_k = \{0, 2, 4, \ldots, 2k, 2k+1\}$ and $\mathcal{L} = \{2\mathbb{N}\} \cup \{L_k \mid k \in \mathbb{N}\}$. Let e such that $W_e = 2\mathbb{N}$ and p using the S-m-n Theorem such that, for all k,

$$W_{p(k)} = L_k.$$

We first show that $\mathcal{L}$ is **TxtSdMonWMonEx**-learnable. We let a learner h such that, for all σ,

$$h(\mathrm{content}(\sigma)) = \begin{cases} e, & \text{if every } x \in \mathrm{content}(\sigma) \text{ is even;} \\ p(y), & \text{if } y \text{ is the least odd datum in } \mathrm{content}(\sigma). \end{cases}$$

Let $L_k \in \mathcal{L}$ and T be a text for L_k. Thus, there is n_0 such that $T(n_0 - 1) = 2k+1$ and any element in $\mathrm{content}(T[n_0 - 1])$ is even. Then, we have, for all $n \geq n_0$, $h(\mathrm{content}(T[n_0])) = h(\mathrm{content}(T[n]))$ and $W_{h(T[n_0])} = W_{p(k)} = L_k$. It is easy to see that h makes exactly one mind change on T and this is at n_0. We have $W_e \cap \mathrm{content}(T)$ is a subset of $W_{p(k)} \cap \mathrm{content}(T)$ as $\{0, 2, \ldots, 2k\} \subseteq L_k$. Thus h is monotone. Additionally, h is weakly monotone as it change its mind only at the first time an odd element is presented in the text, and all previous hypotheses are for $2\mathbb{N}$.

Now suppose that there are $h' \in \mathcal{R}$ and h' **TxtGSMonEx**-learns $\mathcal{L}$. Let σ be a locking sequence of h' on $2\mathbb{N}$ and let k be such that, for all $x \in \mathrm{content}(\sigma)$, $x \leq 2k+1$. We let T be a text for L_k starting with σ. As $2\mathbb{N} \not\subseteq L_k$ we have that h' is not strongly monotone on T or h does not **TxtGEx**-learn L_k from T. □

Theorem 7.2. *There is $\mathcal{L}$ such that $\mathcal{L}$ is* **TxtSdWMonEx**-*learnable but not* **TxtGMonEx**-*learnable.*

Proof. This is a standard proof which we include for completeness. Let $L_k = \{x \mid x \leq 2k+1\}$ and $\mathcal{L} = \{2\mathbb{N}\} \cup \{L_k \mid k \in \mathbb{N}\}$. Let e such that $W_e = 2\mathbb{N}$ and p using the S-m-n Theorem such that, for all k,

$$W_{p(k)} = L_k.$$

We define a learner h such that, for all σ,

$$h(\mathrm{content}(\sigma)) = \begin{cases} e, & \text{if every element in } \mathrm{content}(\sigma) \text{ is even;} \\ p(y), & \text{else, } y \text{ is the maximal odd element in } \mathrm{content}(\sigma). \end{cases}$$

Let $L_k \in \mathcal{L}$ and a T be a text for L_k. Then there is n_0 such that $2k+1 \in \mathrm{content}(T[n_0])$ for the first time. Thus we have that, for all $n \geq n_0, h(\mathrm{content}(T[n_0])) = h(\mathrm{content}(T[n]))$ and $W_{h(\mathrm{content}(T[n_0]))} = W_{p(k)} = L_k$. Obviously, h learns L_k weakly monotonically as the learner only change its mind if a greater odd element appears in the text.

Suppose now there is a learner $h' \in \mathcal{R}$ such that h' **TxtGMonEx**-learns $\mathcal{L}$. Let σ be a locking sequence of h' on $2\mathbb{N}$ and let k be such that, for all $x \in \mathrm{content}(\sigma)$, $x \leq 2k+1$. Let $\sigma' \supseteq \sigma$ be a locking sequence of h' on L_k. Let $\sigma'' \supseteq \sigma'$ be a locking sequence of h' on L_{k+1} and let T be a text for L_{k+1} starting with σ''. Then, we have

$$W_{h'(\sigma)} = 2\mathbb{N};$$
$$W_{h'(\sigma')} = L_k;$$
$$W_{h'(\sigma'')} = L_{k+1}.$$

As the datum $2k+2$ is in $2\mathbb{N}$ and in L_{k+1} but not in L_k, h' is not monotone on the text T for L_{k+1}. □

The following theorem is an extension of a theorem from [3], where the theorem has been shown for decisive learning instead of strongly decisive learning.

Theorem 7.3. *Let $\mathbb{N} \in \mathcal{L}$ and $\mathcal{L}$ be* **TxtGEx**-*learnable. Then, we have $\mathcal{L}$ is* **TxtGSDecEx**-*learnable.*

Proof. In this proof, for any two given sets A, B, we write $A =^* B$ iff $(A \setminus B) \cup (B \setminus A)$ is finite, i.e. if A and B are finite variants of one another. Let h be a learner in Fulk normal form such that h **TxtGEx**-learns $\mathcal{L}$ with $\mathbb{N} \in \mathcal{L}$. As h is strongly locking on $\mathcal{L}$ there is a locking sequence of h on $\mathbb{N}$. Using this locking sequence we get an uniformly enumerable sequence $(L_i)_{i \in \mathbb{N}}$ of languages such that,

1. for all $i \neq j$, $L \supseteq L_i$, $L' \supseteq L_j$ with $L_i =^* L$, $L_j =^* L'$, we have $L \neq L'$;
2. for all $L \supseteq L_i$ with $L_i =^* L$, we have $L \notin \mathcal{L}$.

For every σ we define a set $N(\sigma)$ such that, for every σ,

$$N(\sigma) = L_{|\sigma|} \cup \mathrm{content}(\sigma).$$

Furthermore, with the S-m-n Theorem there is $r \in \mathcal{R}$ such that

$$\forall \sigma : W_{r(\sigma)} = N(\sigma).$$

We let M be total computable such that, for all σ, $M(\sigma)$ is the finite set which includes any sequence $\tau \subseteq \sigma$ such that

- for all $x \in \text{content}(\tau)$, $\Phi_{h(\tau)}(x) \le |\sigma|$; and
- for all $\tau' \supseteq \tau$ with $\text{content}(\tau') \subseteq \text{content}(\tau)$ and $\left|\tau'\right| \le |\sigma|$, $h(\tau) = h(\tau')$.

Using the S-m-n Theorem we get a function $p \in \mathcal{R}$ such that, for all σ,

$$W_{p(\sigma)} = \bigcup_{t \in \mathbb{N}} \begin{cases} W^t_{h(\sigma)}, & \text{if } \forall \rho \in W^t_{h(\sigma)} : h(\sigma) = h(\sigma \diamond \rho); \\ N(\sigma), & \text{otherwise.} \end{cases}$$

We will use the $p(\sigma)$ as hypotheses. Note that any hypothesis $p(\sigma)$ is either semantically equivalent to $h(\sigma)$ or, if σ is not a locking sequence of h for any language, $p(\sigma)$ is an index for a finite superset of $L_{|\sigma|}$. In the latter case we call the hypothesis $p(\sigma)$ *poisoned*. We will also sometimes default to $r(\sigma)$ as hypothesis, which we will also consider a poisoned hypothesis.

We define a learner h' such that, for all σ,

$$h'(\sigma) = \begin{cases} p(\min(M(\sigma))), & \text{if } M(\sigma) \neq \emptyset; \\ r(\sigma), & \text{otherwise.} \end{cases}$$

Let $L \in \mathcal{L}$ and T be a text for L. As h is strongly locking and h **TxtGEx**-learns $\mathcal{L}$ there is a minimal n_0 such that, for all $\sigma \in \mathbb{S}\text{eq}(L)$, $h(T[n_0]) = h(T[n_0] \diamond \sigma)$ and $W_{h(T[n_0])} = L$. Thus, there is $n_1 > n_0$ such that, for all $x \in \text{content}(T[n_0])$, $\Phi_{h(T[n_0])}(x) \le n_1$. This implies that, for all $n \ge n_1$, $h'(T[n_1]) = h'(T[n])$ and

$$W_{h'(T[n_1])} = W_{p(\min(M(T[n_1])))} = \bigcup_{t \in \mathbb{N}} W^t_{h(T[n_0])} = L.$$

Next, we will show that h' is strongly decisive. Suppose there are $i \le j \le k$ such that $W_{h'(T[i])} = W_{h'(T[k])}$ and $h'(T[i]) \neq h'(T[j])$.

Case 1: $h'(T[i])$ is *not* a poisoned hypothesis. Let τ_i, τ_j, τ_k be such that $h'(T[i]) = p(\tau_i)$, $h'(T[j]) \in \{r(\tau_j), p(\tau_j)\}$ and $h'(T[k]) \in \{r(\tau_k), p(\tau_k)\}$. We will show that $\text{content}(\tau_j) \subseteq W_{h'(T[k])}$ and $\text{content}(\tau_j) \nsubseteq W_{h'(T[i])}$, which implies $W_{h'(T[i])} \neq W_{h'(T[k])}$, our desired contradiction.

We have $\tau_i \subset \tau_j \subseteq \tau_k$, from the construction of h' and using $h'(T[i]) \neq h'(T[j])$. From the construction of M (and the definition of $N(\sigma)$) we have $\text{content}(\tau_k) \subseteq W_{h'(T[k])}$; in particular,

$$\text{content}(\tau_j) \subseteq \text{content}(\tau_k) \subseteq W_{h'(T[k])}.$$

Recall now that h made a mind change between τ_i and τ_j, so since $h'(T[i])$ is *not* a poisoned hypothesis, $W_{h'(T[i])}$ cannot contain all of $\text{content}(\tau_j)$, showing $\text{content}(\tau_j) \nsubseteq W_{h'(T[i])}$ as desired.

Case 2: $h'(T[i])$ *is* poisoned. Thus, we have $\text{content}(T[i]) \subseteq W_{h'(T[i])}$ form the definition of $N(\sigma)$.

Case 2.1: $h'(T[k])$ is *not* poisoned. Thus, $T[k]$ is a locking sequence on h for a language $L \in \mathbf{TxtGEx}(h)$ and $W_{h'(T[k])} \in \mathbf{TxtGEx}(h)$. As $h'(T[i])$ is poisoned we have $W_{h'(T[i])} \notin \mathbf{TxtGEx}(h)$. Thus, we get $W_{h'(T[i])} \neq W_{h'(T[k])}$, a contradiction.

Case 2.2: $h'(T[k])$ *is* poisoned. As $i < k$ we have

$$W_{h'(T[i])} =^* N(T[i]) =^* L_i \neq^* L_k =^* N(T[k]) =^* W_{h'(T[k])}.$$

Thus, $W_{h'(T[i])} \neq W_{h'(T[k])}$, a contradiction. □

Theorem 7.4. *We have that any monotone* **TxtGEx***-learnable class of languages is strongly decisive learnable, while the converse does not hold, i.e.*

$$[\mathbf{TxtGMonEx}] \subset [\mathbf{TxtGSDecEx}].$$

Proof. Let $h \in \mathcal{R}$ be a learner and $\mathcal{L} = \mathbf{TxtGMonEx}(h)$. We distinguish the following two cases. We call $\mathcal{L}$ *dense* iff it contains a superset of every finite set.

Case 1: $\mathcal{L}$ is dense. We will show now that h **TxtGSMonEx**-learns the class $\mathcal{L}$. Let $L \in \mathcal{L}$ and T be a text for L. Suppose there are i and j with $i < j$ such that $W_{h(T[i])} \nsubseteq W_{h(T[j])}$. Thus, we have $W_{h(T[i])} \setminus W_{h(T[j])} \neq \emptyset$. Let $x \in W_{h(T[i])} \setminus W_{h(T[j])}$. As $\mathcal{L}$ is dense there is a language $L' \in \mathcal{L}$ such that $\text{content}(T[j]) \cup \{x\} \subseteq L'$. Let T' be a text for L' and T'' be such that $T'' = T[j] \diamond T'$. Obviously, T'' is a text for L'. We have that $x \in W_{h(T''[i])}$ but $x \notin W_{h(T''[j])}$ which is a contradiction as h is monotone. Thus, h **TxtGSMonEx**-learns $\mathcal{L}$, which implies that h **TxtGWMonEx**-learns $\mathcal{L}$. Using Corollary 4.6 we get that $\mathcal{L}$ is **TxtGSDecEx**-learnable.

Case 2: $\mathcal{L}$ is not dense. Thus, $\mathcal{L}' = \mathcal{L} \cup \{\mathbb{N}\}$ is **TxtGEx**-learnable. Using Theorem 7.3 $\mathcal{L}'$ is **TxtGSDecEx**-learnable and therefore so is $\mathcal{L}$.

Regarding the inclusion being proper, recall from Corollary 4.6 that [**TxtGWMonEx**] $\subset$ [**TxtGSDecEx**]. Let $\mathcal{L} \in$ [**TxtGSDecEx**] $\setminus$ [**TxtGWMonEx**]. As seen in *Case 2*, we can assume, without loss of generality, that $\mathcal{L}$ is dense (by adding $\mathbb{N}$ if necessary). If $\mathcal{L}$ was in [**TxtGMonEx**] then it would be in [**TxtGSMonEx**] (as shown in *Case 1*) and thus in [**TxtGWMonEx**], in contrast to what we supposed. Thus we get $\mathcal{L} \in$ [**TxtGSDecEx**] $\setminus$ [**TxtGMonEx**] as desired. □

Acknowledgements

The authors would like to thank the anonymous reviewers for the conference version, which helped improve the current version. Furthermore we would like to thank the anonymous reviewers of the journal versions which made big efforts which greatly benefited this paper.

References

[1] D. Angluin, Inductive inference of formal languages from positive data, Inf. Control 45 (1980) 117–135.
[2] L. Blum, M. Blum, Toward a mathematical theory of inductive inference, Inf. Control 28 (1975) 125–155.
[3] G. Baliga, J. Case, W. Merkle, F. Stephan, W. Wiehagen, When unlearning helps, Inform. and Comput. 206 (2008) 694–709.
[4] J. Case, Periodicity in generations of automata, Math. Syst. Theory 8 (1974) 15–32.
[5] J. Case, Infinitary self-reference in learning theory, J. Exp. Theor. Artif. Intell. 6 (1994) 3–16.
[6] J. Case, T. Kötzing, Strongly non-U-shaped learning results by general techniques, in: Proc. of COLT (Conference on Learning Theory), 2010, pp. 181–193.
[7] J. Case, S. Moelius, Optimal language learning from positive data, Inform. and Comput. 209 (2011) 1293–1311.
[8] M. Fulk, Prudence and other conditions on formal language learning, Inform. and Comput. 85 (1990) 1–11.
[9] E. Gold, Language identification in the limit, Inf. Control 10 (1967) 447–474.
[10] K.P. Jantke, Monotonic and non-monotonic inductive inference of functions and patterns, in: J. Dix, K.P. Jantke, P.H. Schmitt (Eds.), Nonmonotonic and Inductive Logic, in: Lecture Notes in Computer Science, vol. 543, Springer-Verlag, Berlin, 1991, pp. 161–177.
[11] S. Jain, D. Osherson, J. Royer, A. Sharma, Systems that Learn: An Introduction to Learning Theory, second edition, MIT Press, Cambridge, MA, 1999.
[12] S. Jain, A. Sharma, Generalization and specialization strategies for learning r.e. languages, Ann. Math. Artif. Intell. 23 (1998) 1–26.
[13] T. 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.
[14] T. Kötzing, A solution to Wiehagen's thesis, in: Proc. of STACS (Symposium on Theoretical Aspects of Computer Science), 2014, pp. 494–505.
[15] T. Kötzing, R. Palenta, A map of update constraints in inductive inference, in: Peter Auer, Alexander Clark, Thomas Zeugmann, Sandra Zilles (Eds.), Algorithmic Learning Theory, 25th International Conference, ALT 2014, Bled, Slovenia, October 8–10, 2014, in: Lecture Notes in Artificial Intelligence, vol. 8776, Springer, Berlin, Heidelberg, New York, 2014, pp. 40–54. Proceedings.
[16] E. Kinber, F. Stephan, Language learning from texts: mind changes, limited memory and monotonicity, Inform. and Comput. 123 (1995) 224–241.
[17] S. Lange, T. Zeugmann, Monotonic versus non-monotonic language learning, in: G. Brewka, K.P. Jantke, P.H. Schmitt (Eds.), Nonmonotonic and Inductive Logic, Second International Workshop, Reinhardsbrunn Castle, Germany, December 1991, in: Lecture Notes in Artificial Intelligence, vol. 659, Springer-Verlag, Berlin, Heidelberg, 1993, pp. 254–269.
[18] D. Osherson, M. Stob, S. Weinstein, Learning strategies, Inf. Control 53 (1982) 32–51.
[19] D. Osherson, M. Stob, S. Weinstein, Systems that Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists, MIT Press, Cambridge, MA, 1986.
[20] J. Royer, J. Case, Subrecursive Programming Systems: Complexity and Succinctness, Research Monograph in Progress in Theoretical Computer Science, Birkhäuser, Boston, 1994.
[21] H. Rogers, Theory of Recursive Functions and Effective Computability, McGraw-Hill, New York, 1967, Reprinted by MIT Press, Cambridge, MA, 1987.
[22] G. Schäfer-Richter, Über Eingabeabhängigkeit und Komplexität von Inferenzstrategien, PhD thesis, RWTH Aachen, 1984.
[23] K. Wexler, P. Culicover, Formal Principles of Language Acquisition, MIT Press, Cambridge, MA, 1980.
[24] R. Wiehagen, A thesis in inductive inference, in: J. Dix, K.P. Jantke, P.H. Schmitt (Eds.), Nonmonotonic and Inductive Logic, in: Lecture Notes in Computer Science, vol. 543, Springer-Verlag, Berlin, Heidelberg, 1991, pp. 184–207.