LEARNING LANGUAGES IN THE LIMIT FROM POSITIVE INFORMATION
WITH FINITELY MANY MEMORY CHANGES

TIMO KÖTZING, KAREN SEIDEL

Abstract. We investigate learning collections of languages from texts by an inductive inference machine with access to the current datum and a bounded memory in form of states. Such a bounded memory states (BMS) learner is considered successful in case it eventually settles on a correct hypothesis while exploiting only finitely many different states.

We give the complete map of all pairwise relations for an established collection of criteria of successful learning. Most prominently, we show that non-U-shapedness is not restrictive, while conservativeness and (strong) monotonicity are. Some results carry over from iterative learning by a general lemma showing that, for a wealth of restrictions (the semantic restrictions), iterative and bounded memory states learning are equivalent. We also give an example of a non-semantic restriction (strongly non-U-shapedness) where the two settings differ.

1. Introduction

We are interested in 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; this is sometimes called inductive inference, a branch of (algorithmic) learning theory. For example, a learner $M$ might be presented more and more even numbers. After each new number, $M$ outputs a description for a language as its conjecture. The learner $M$ might decide to output a program for the set of all multiples of $4$, as long as all numbers presented are divisible by $4$. Later, when $M$ sees an even number not divisible by $4$, it might change this guess to a program for the set of all multiples of $2$.

Many criteria for deciding whether a learner $M$ is successful on a language $L$ have been proposed in the literature. Gold, in his seminal paper [Gol67], gave a first, simple learning criterion, $\text{TxtEx}$-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 an $\text{TxtEx}$-learner (this learner constantly outputs a description for $L$). Thus, we are interested in analyzing for which classes of languages $L$ there is a single learner $M$ learning each member of $L$. Sometimes, this framework is called language learning in the limit and has been studied extensively, using a wide range of learning criteria similar to $\text{TxtEx}$-learning (see, for example, the textbook [JOR99]).

One major criticism of the model suggested by Gold is its excessive use of memory, see for example [CM08]: for each new hypothesis the entire history of past data is available. Iterative learning is the most common variant of learning in the limit which addresses memory constraints: the memory of the learner on past data is just its current hypothesis. Due to the padding lemma [JOR99], this memory is not necessarily void, but only finitely many data can be memorized in the hypothesis. There is a comprehensive body of work on iterative learning, see, e.g., [CK10, CM08, JKMS16, JMZ13, JORS99].

Another way of modelling restricted memory learning is to grant the learner access to not their current hypothesis, but a state which can be used in the computation of the next hypothesis (and next state). This was introduced in [CCJS07] and called bounded memory states (BMS) learning. It is a reasonable assumption to have a countable reservoir of states. Assuming a computable enumeration of these states, we use natural numbers to refer to them. Note that allowing arbitrary use of all natural numbers as states would effectively allow a learner to store all seen data in the state, thus giving the same mode as Gold’s original setting. Probably the minimal way to restrict the use of states is to demand for successful learning that a learner

---

$^1$\text{Txt} stands for learning from a text of positive examples; \text{Ex} stands for explanatory.
must stop using new states eventually (but may still traverse among the finitely many states produced so far, and may use infinitely many states on data for a non-target language). It was claimed that this setting is equivalent to iterative learning [CCJS07, Remark 38] (this restriction is called ClassBMS there, we refer to it by TxtBMS,Ex). However, this was only remarked for the plain setting of explanatory learning; for further restrictions, the setting is completely unknown, only for explicit constant state bounds a few scattered results are known, see [CCJS07, CK13].

In this paper, we consider a wealth of restrictions, described in detail in Section 2 (after an introduction to the general notation of this paper). Following the approach of giving maps of pairwise relations suggested in [KS16], we give a complete map in Figure 1. We note that this map is the same as the map for iterative learning given in [JKMS16], but partially for different reasons.

In Lemma 4.1, we show that, for many restrictions (the so-called semantic restrictions, where only the semantics of hypotheses are restricted) the learning setting with bounded memory states is equivalent to learning iteratively. This proves and generalizes the aforementioned remark in [CCJS07] to a wide class of restrictions. The iterative learner uses the hypotheses of the BMSs-learner on an equivalent text and additionally pads a subgraph of the translation diagram to it. It keeps track of all states visited so far together with the datum which caused the first transfer to the respective state. This way we can reconstruct the last first-time-visited state while observing the equivalent text sequence. Moreover, the equivalent text prevents the iterative learner from returning to a previously visited state but the last one and hence enables the required convergence.

However, if restrictions are not semantic, then iterative and bounded memory states learning can differ. We show this concretely for the case of so-called strongly non-U-shaped learning in Theorem 4.3. Inspired by cognitive science research [SS82, MPU+92], a semantic version of this requirement was defined in [BCM*08] and later the syntactic variant was introduced in [CM11]. Both requirements have been extensively studied, see [CCI13] for a survey and moreover [CK13], [CK16], [KSS17]. The proof of Theorem 4.3 uses an intricate ORT-argument, which might suggest that the two settings, while different, are very similar nonetheless. It is based on the proof that strong non-U-shapedness restricts BMSs-Ex-learning. The proof of the latter result combines the techniques for showing that strong non-U-shapedness restricts iterative learning, as proved in [CK13, Theorem 5.7], and that not every class strongly monotonically learnable by an iterative learner is strongly non-U-shapedly learnable by an iterative learner, see [JKMS16, Theorem 5]. Moreover, it relies on showing that state decisiveness can be assumed in Lemma 4.1.

The remainder of Section 4 completes the map given in Figure 1 for the case of syntactic restrictions (since these do not carry over from the setting of iterative learning). All syntactic learning requirements are closely related to strongly locking learners. The fundamental concept of a locking sequence was introduced by [BB75]. For a similar purpose than ours [JKMS16] introduced strongly locking learners. We generalize their construction for certain syntactically restricted iterative learners from a strongly locking iterative learner. Finally, we obtain that all non-semantic learning restrictions also coincide for BMSs-learning.

2. LEARNERS, SUCCESS CRITERIA AND OTHER TERMINOLOGY

As far as possible, we follow [JORS99] on the learning theoretic side and [Od99] for computability theory. We recall the most essential notation and definitions.

We let \( \mathbb{N} \) denote the natural numbers including 0. For a function \( f \) we write \( \text{dom}(f) \) for its domain and \( \text{ran}(f) \) for its range. If we deal with (a subset of) a cartesian product, we are going to refer to the projection functions to the first or second coordinate by \( \text{pr}_1 \) and \( \text{pr}_2 \), respectively.

Further, \( X^{<\omega} \) denotes the finite sequences over the set \( X \) and \( X^\omega \) stands for the countably infinite sequences over \( X \). For every \( \sigma \in X^{<\omega} \) and \( t \leq |\sigma| \), \( t \in \mathbb{N} \), we let \( \sigma[t] := \{(s, \sigma(s)) \mid s < t\} \) denote the restriction of \( \sigma \) to \( t \). Moreover, for sequences \( \sigma, \tau \in X^{<\omega} \) their concatenation is denoted by \( \sigma \tau \). Finally, we write last(\( \sigma \)) for the last element of \( \sigma \), \( \sigma(|\sigma| - 1) \), and \( \sigma^- \) for the initial segment of \( \sigma \) without last(\( \sigma \)), i.e. \( \sigma[|\sigma| - 1] \). Clearly, \( \sigma = \sigma^- \text{last}(\sigma) \).

For a finite set \( D \subseteq \mathbb{N} \) and a finite sequence \( \sigma \in X^{<\omega} \), we denote by \( \langle D \rangle \) and \( \langle \sigma \rangle \) a canonical index for \( D \) or \( \sigma \), respectively. Further, we fix a Goedel pairing function \( \langle \cdot, \cdot \rangle \) with two arguments.
Let $L \subseteq \mathbb{N}$. We interpret every $n \in \mathbb{N}$ as a code for a word. If $L$ is recursively enumerable, we call $L$ a language.

We fix a programming system $\varphi$ as introduced in [RC94]. Briefly, in the $\varphi$-system, for a natural number $p$, we denote by $\varphi_p$ the partial computable function with program code $p$. We call $p$ an index for $W_p$ defined as $\operatorname{dom}(\varphi_p)$.

In reference to a Blum complexity measure $\Phi_p$, for all $p, t \in \mathbb{N}$, we denote by $W_p^t \subseteq W_p$ the recursive set of all natural numbers less or equal to $t$, on which the machine executing $p$ halts in at most $t$ steps, i.e.

$$W_p^t = \{ x \mid x \leq t \land \Phi_p(x) \leq t \}.$$ Moreover, the well-known s-m-n theorem gives finite and infinite recursion theorems, see [Cas74], [Cas94], [Osi99]. We will refer to Case’s Operator Recursion Theorem ORT in its 1-1-form, for example [Kot09].

Throughout the paper, we let $\Sigma = \mathbb{N} \cup \{ \# \}$ be the input alphabet with $n \in \mathbb{N}$ interpreted as code for a word in the language and $\#$ interpreted as pause symbol, i.e. no new information. Further, let $\Omega = \mathbb{N} \cup \{ ? \}$ be the output alphabet with $p \in \mathbb{N}$ interpreted as $\varphi$-index and $?$ as no hypothesis or repetition of the last hypothesis, if existent. A function with range $\Omega$ is called a hypothesis generating function.

A learner is always a (partial) computable function

$$M : \operatorname{dom}(M) \subseteq \Sigma^\omega \to \Omega.$$ The set of all total computable functions $M : \Sigma^\omega \to \Omega$ is denoted by $\mathcal{R}$.

Let $f \in \Sigma^\omega \cup \Sigma^*$, then the content of $f$, defined as $\operatorname{content}(f) := \operatorname{ran}(f) \backslash \{ \# \}$, is the set of all natural numbers, about which $f$ gives some positive information. The set of all tests for the language $L$ is defined as

$$\operatorname{Txt}(L) := \{ T \in \Sigma^\omega \mid \operatorname{content}(T) = L \}.$$ Definition 2.1. Let $M$ be a learner. $M$ is an iterative learner or it-learner, for short $M \in \textbf{It}$, if there is a computable (partial) hypothesis generating function $h_M : \Omega \times \Sigma \to \Omega$ such that $M = h_M^{\dagger}$ where $h_M^{\dagger}$ is defined on finite sequences by

$$h_M^{\dagger}(\epsilon) = ?; \quad h_M^{\dagger}(h_M^{\dagger}(\sigma), x).$$

Definition 2.2. Let $M$ be a learner. $M$ is a bounded memory states learner or BMS-learner, for short $M \in \textbf{BMS}$, if there are a computable (partial) hypothesis generating function $h_M : \mathbb{N} \times \Sigma \to \Omega$ and a computable (partial) state transition function $s_M : \mathbb{N} \times \Sigma \to \mathbb{N}$ such that $\operatorname{dom}(h_M) = \operatorname{dom}(s_M)$ and $M = h_M^\ast$ where $h_M^\ast$ and $s_M^\ast$ are defined on finite sequences by

$$s_M^\ast(\epsilon) = 0; \quad h_M^\ast(s_M^\ast(\sigma), x); \quad s_M^\ast(\sigma, x) = s_M(s_M^\ast(\sigma), x).$$

Note that every iterative learner gives a BMS-learner by identifying the hypothesis space $\Omega$ with the set of states via a computable bijection between $\mathbb{N}$ and $\Omega$. The resulting BMS-learner will succeed on the same languages the iterative learner does learn. Further, as the set of visited states contains exactly all hypotheses the learner puts out, this BMS-learner only uses finitely many states on all texts for languages it explanatory learns. In [CCJS07] Rem. 38 it is claimed that $\textbf{BMS}^\ast$-learners and iterative learners are equally powerful on texts. This also follows from our more general Lemma 3.1. The above intuition is formalized in the corresponding proof.

Definition 2.2 may be stated more generally for arbitrary finite or infinite sets of states $Q$, instead of $\mathbb{N}$. Moreover, $s_M^\ast$ and $h_M^\ast$ can easily be generalized to functions taking also a starting state $s$ as input by

$$s_M^\ast(s, \epsilon) = s; \quad h_M^\ast(s, \sigma, x) = h_M(s_M^\ast(s, \sigma), x);$$
We now clarify what we mean by successful learning.

**Definition 2.3.** Let $M$ be a learner and $\mathcal{L}$ a collection of languages.

1. Let $L \in \mathcal{L}$ be a language and $T \in \text{Txt}(L)$ a text for $L$ presented to $M$.
   (a) We call $h = (h_t)_{t \in \mathbb{N}}$ where $h_t := M(T[t])$ for all $t \in \mathbb{N}$, the **learning sequence of $M$ on $T$**.
   (b) $M$ learns $L$ from $T$ in the limit, for short $M$ **Ex-learns $L$ from $T$** or **Ex$(M,T)$**, if there exists $t_0 \in \mathbb{N}$ such that $W_{h_{t_0}} = \text{content}(T)$ and $\forall t \geq t_0 \ (h_t \neq ? \Rightarrow h_t = h_{t_0})$.

2. $M$ learns $\mathcal{L}$ in the limit, for short $M$ **Ex-learns $\mathcal{L}$**, if **Ex$(M,T)$** for every $L \in \mathcal{L}$ and every $T \in \text{Txt}(L)$.

**Definition 2.4.** Let $\mathcal{L}$ be a collection of languages. $\mathcal{L}$ is learnable in the limit or **Ex-learnable**, if there exists a learner $M$ that **Ex-learns $\mathcal{L}$**.

**Ex**-learning is the most common definition for successful learning in inductive inference and corresponds to the notion of identifiability in the limit by [Gol67], where the learner eventually decides on one correct hypotheses.

In our investigations, the most important additional requirement on a successful learning process for a **BMS**-learner is to use finitely many states only, as stated in the following definition.

**Definition 2.5.** Let $M$ be a **BMS**-learner and $T \in \text{Txt}$. We say that $M$ uses finitely many memory states on $T$, for short **BMS$_s$(M,T)**, if $\{s^*_{M}(T[t]) \mid t \in \mathbb{N}\}$ is finite.

We list the most common additional requirements regarding the learning sequence, which may tag a learning process. For this we first recall the notion of consistency of a sequence with a set.

**Definition 2.6.** Let $f \in \Sigma^{<\omega} \cup \Sigma^\omega$ and $A \subseteq \Sigma$. We define

$$\text{Cons}(f,A) \iff \text{content}(f) \subseteq A$$

and say $f$ is consistent with $A$.

The listed properties of the learning sequence have been at the center of different investigations. Studying how they relate to one another did begin in [KP16], [KS16], [JKMS16] and [AKS18].

**Definition 2.7.** Let $M$ be a learner, $T \in \text{Txt}$ and $h = (h_t)_{t \in \mathbb{N}} \in \Omega^\omega$ the learning sequence of $M$ on $T$, i.e. $h_t = M(T[t])$ for all $t \in \mathbb{N}$. We write

1. $\text{Conv}(M,T)$ ([Ang80]), if $M$ is conservative on $T$, i.e., for all $s,t$ with $s \leq t$ holds
   \[\text{Cons}(T[t],W_{h_t}) \Rightarrow h_s = h_t.\]
2. $\text{Dec}(M,T)$ ([OSW82]), if $M$ is decisive on $T$, i.e., for all $r,s,t$ with $r \leq s \leq t$ holds
   \[W_{h_r} = W_{h_t} \Rightarrow W_{h_s} = W_{h_t}.\]
3. $\text{Caut}(M,T)$ ([OSW86]), if $M$ is cautious on $T$, i.e., for all $s,t$ with $s \leq t$ holds
   \[\neg W_{h_s} \subseteq W_{h_t}.\]
4. $\text{WMon}(M,T)$ ([Jan91], [Wie91]), if $M$ is weakly monotonic on $T$, i.e., for all $s,t$ with $s \leq t$ holds
   \[\text{Cons}(T[t],W_{h_t}) \Rightarrow W_{h_s} \subseteq W_{h_t}.\]
5. $\text{SMon}(M,T)$ ([Jan91], [Wie91]), if $M$ is strongly monotonic on $T$, i.e., for all $s,t$ with $s \leq t$ holds
   \[W_{h_s} \cap \text{content}(T) \subseteq W_{h_t} \cap \text{pos}(T).\]
6. $\text{NU}(M,T)$ ([BCM+08]), if $M$ is non-U-shaped on $T$, i.e., for all $r,s,t$ with $r \leq s \leq t$ holds
   \[W_{h_r} = W_{h_s} = \text{content}(T) \Rightarrow W_{h_t} = W_{h_s}.\]
7. $\text{SNU}(M,T)$ ([CM11]), if $M$ is strongly non-U-shaped on $T$, i.e., for all $r,s,t$ with $r \leq s \leq t$ holds
   \[W_{h_r} = W_{h_s} = \text{content}(T) \Rightarrow h_r = h_s.\]
8. $\text{SDec}(M,T)$ ([KP16]), if $M$ is strongly decisive on $T$, i.e., for all $r,s,t$ with $r \leq s \leq t$ holds
   \[W_{h_r} = W_{h_t} \Rightarrow h_r = h_s.\]
9. $\text{WB}(M,T)$ ([KS16]), if $M$ is witness-based on $T$, i.e., for all $r,t$ such that for some $s$ with $r < s \leq t$ holds $h_r \neq h_s$ we have $\text{content}(T[s]) \cap (W_{h_t} \setminus W_{h_r}) \neq \emptyset$. 

\[s^*_M(s,\sigma^r x) = s^*_M(s,\sigma), x)\].
It is easy to see that \( \text{Conv}(M, T) \) implies \( \text{SNU}(M, T) \) and \( \text{WMon}(M, T) \); \( \text{SDec}(M, T) \) implies \( \text{Dec}(M, T) \) and \( \text{SNU}(M, T) \); \( \text{SMon}(M, T) \) implies \( \text{Caut}(M, T) \), \( \text{Dec}(M, T) \), \( \text{Mon}(M, T) \), \( \text{WMon}(M, T) \) and finally \( \text{Dec}(M, T) \), \( \text{WMon}(M, T) \) and \( \text{SNU}(M, T) \) imply \( \text{NU}(M, T) \). Figure 1 includes the resulting backbone with arrows indicating the aforementioned implications. Further, \( \text{Wb}(M, T) \) implies \( \text{Conv}(M, T) \), \( \text{SDec}(M, T) \) and \( \text{Caut}(M, T) \).

In order to characterize what successful learning means, these predicates may be combined with the explanatory convergence criterion. For this, we let \( \Delta := \{ \text{Caut}, \text{Conv}, \text{Dec}, \text{SDec}, \text{WMon}, \text{Mon}, \text{SMon}, \text{NU}, \text{SNU}, \text{T} \} \) denote the set of admissible learning restrictions, with \( \text{T} \) standing for no restriction. Further, a learning success criterion is a predicate being the intersection of the convergence criterion \( \text{Ex} \) with arbitrarily many admissible learning restrictions. This means that the sequence of hypotheses has to converge and in addition has the desired properties. Therefore, the collection of all learning success criteria is

\[
\left\{ \bigcap_{i=0}^{n} \delta_i \cap \text{Ex} \mid n \in \mathbb{N}, \forall i \leq n(\delta_i \in \Delta) \right\}.
\]

Note that plain explanatory convergence is a learning success criterion by letting \( n = 0 \) and \( \delta_0 = \text{T} \).

We refer to all \( \delta \in \{ \text{Caut}, \text{Cons}, \text{Dec}, \text{Mon}, \text{SMon}, \text{WMon}, \text{NU}, \text{T} \} \) also as semantic learning restrictions, as they do not require the learner to settle on exactly one hypothesis.

In order to state observations about how two ways of defining learning success relate to each other, the learning power of the different settings is encapsulated in notions \([\alpha\text{Txt}\beta]\) defined as follows.

**Definition 2.8.** Let \( \alpha \) be a property of partial computable functions from the set \( \Sigma^{<\omega} \) to \( \mathbb{N} \) and \( \beta \) a learning success criterion. We denote by \([\alpha\text{Txt}\beta]\) the set of all collections of languages that are \( \beta \)-learnable from texts by a learner \( M \) with the property \( \alpha \).

At position \( \alpha \), we restrict the set of admissible learners for example by requiring them to be iterative or finite bounded memory states learners. The properties stated at position \( \alpha \) are independent of learning success. In contrast, at position \( \beta \), the required learning behavior and convergence criterion are specified. We do not use separators in the notation to stay consistent with established notation in the field that was inspired by \([\text{JORS}99]\).

For example, a collection of languages \( \mathcal{L} \) lies in \([\text{BMSTxtBMSTxt}\text{ConvEx}]\) if and only if there is a bounded memory states learner \( M \) conservatively explanatory learning every \( L \in \mathcal{L} \) from texts while using only finite memory. More concretely, for all \( L \in \mathcal{L} \) and for every text \( T \in \text{Txt}(L) \) we have \( \text{Conv}(M, T), \text{BMSTxtBMSTxt}(M, T) \) and \( \text{Ex}(M, T) \).

The proofs of Lemmata 3.1 and 4.1 employ the following property of learning requirements and learning success criteria, that applies to all such considered in this paper.

**Definition 2.9.** Denote the set of all unbounded and non-decreasing functions by \( \mathcal{S} \), i.e.,

\[
\mathcal{S} := \{ s : \mathbb{N} \to \mathbb{N} \mid \forall x \in \mathbb{N} \exists t \in \mathbb{N} : s(t) \geq x \land \forall t \in \mathbb{N} : s(t+1) \geq s(t) \}.
\]

Then every \( s \in \mathcal{S} \) is a so called admissible simulating function.

A predicate \( \beta \) on pairs of learners and texts allows for simulation on equivalent text, if for all simulating functions \( s \in \mathcal{S} \), all texts \( T, T' \in \text{Txt} \) and all learners \( M, M' \) holds: Whenever we have \( \text{content}(T'[t]) = \text{content}(T[s(t)]) \) and \( M'(T'[t]) = M(T[s(t)]) \) for all \( t \in \mathbb{N} \), from \( \beta(M, T) \) we can conclude \( \beta(M', T') \).

Intuitively, as long as the learner \( M' \) conjectures \( h'_t = h_{s(t)} = M(T[s(t)]) \) at time \( t \) and has, in form of \( T'[t] \), the same data available as was used by \( M \) for this hypothesis, \( M' \) on \( T' \) is considered to be a simulation of \( M \) on \( T \).

It is easy to see that all learning success criteria considered in this paper allow for simulation on equivalent text.
### 3. Relations between Semantic Learning Requirements

We show that bounded memory state learners and iterative learners have equal learning power, when a semantic learning requirement is added to the standard convergence criterion. With this the results from iterative learning are transferred to this setting.

The following lemma formally establishes the equal learning power of iterative and BMS\(_{\sigma}\)-learning for all learning success criteria but Conv, SDec and SNU. We are going to prove in Section 4 that even for the three aforementioned non-semantic additional requirements we obtain the same behavior.

**Lemma 3.1.** Let \(\delta\) allow for simulation on equivalent text.

1. We have \([\text{TxtBMS}_{\sigma}\delta\text{Ex}] \supseteq [\text{ItTxt}\delta\text{Ex}].\)
2. If \(\delta\) is semantic then \([\text{TxtBMS}_{\sigma}\delta\text{Ex}] = [\text{ItTxt}\delta\text{Ex}].\)

**Proof.** While \([1]\) and \(\subseteq\) in \([2]\) are easy to verify by using the hypotheses as states, the other inclusion in \([2]\) is more challenging. The iterative learner constructed from the BMS-learner \(M\) uses the hypotheses of \(M\) on an equivalent text and additionally pads a subgraph of the translation diagram of \(M\) to it.

(1) and \(\subseteq\) of (2). Let \(M\) be an iterative learner, i.e. there is a computable function \(h_M : \Omega \times \Sigma \to \Omega\) with \(M = h_M^1\) where \(h_M^1(\varepsilon) = \varepsilon\) and \(h_M^1(\sigma \cdot x) = h_M(h_M^1(\sigma), x)\) for all \(\sigma \in \Sigma^{<\omega}\) and \(x \in \Sigma\). We show that \(M\) can be obtained as a state driven learner by using the hypotheses also as states. For this, we fix the computable bijection \(\pi : Q \to \Omega\) with computable inverse, defined by \(\pi(0) = ?\) and \(\pi(i) = i - 1\) for all \(i > 0\). Then the learner \(N = h_N^1\) with \((s_N, h_N^1(q, x) = (\pi^{-1}(h_M(\pi(q), x)), h_M(\pi(q), x))\) is as wished because the state corresponds via \(\pi\) directly to the last hypothesis of \(M\) and so the learners \(M\) and \(N\) act identically.

Formally, this follows by an induction showing for every \(\tau \in \Sigma^{<\omega}\) that \(s^*_N(\tau) = \pi^{-1}(M(\tau))\) and moreover if \(|\tau| > 0\) we have \(N(\tau) = M(\tau)\). The claim holds for \(\tau = \varepsilon\), because of \(s^*_N(\varepsilon) = 0 = \pi^{-1}(M(\varepsilon))\). In case there are \(\sigma \in \Sigma^{<\omega}\) and \(x \in \Sigma\) such that \(\tau = \sigma \cdot x\), we may assume \(s^*_N(\sigma) = \pi^{-1}(M(\sigma))\) and obtain

\[
\begin{align*}
N(\tau) & \overset{N = h_N^1}{=} h_N(s^*_N(\sigma), x) \overset{s^*_N(\sigma) = \pi^{-1}(M(\sigma))}{=} s_N(\pi^{-1}(M(\sigma)), x) \overset{\text{Def.} s_N}{=} \pi^{-1}(h_M(\pi(\sigma), x)) \overset{M = h_M^1}{=} \pi^{-1}(M(\tau)) \overset{\text{Def.} h_N}{=} h_M(M(\sigma), x) \overset{M = h_M^1}{=} M(\tau).
\end{align*}
\]

That \(M\) in case of learning success uses only finitely many states follows immediately from the \(\text{Ex}\)-convergence, implying to output only finitely many pairwise distinct hypotheses.

\(\subseteq\) of (2). Let \(\mathcal{L} \in [\text{TxtBMS}_{\sigma}\delta\text{Ex}]\) be witnessed by the learner \(M, i.e., there is \(\langle s_M, h_M : Q \times \Sigma \to Q \times \Omega \rangle\) such that \(M = h_M^\mathcal{L}\). Further, we may assume that for all \(L \in \mathcal{L}\) and \(T \in \text{Txt}(T)\) the set of visited states \(s_M^{\mathcal{L}}\{[T[t] \mid t \in \mathbb{N}]\}\) is finite and \(M \delta\text{Ex}\)-learns \(L\) from \(T\).

Intuitively, the iterative learner \(M^\text{It}\) uses the hypotheses of \(M\) on an equivalent text \(\hat{T}\) and additionally pads a subgraph \(V(\sigma)\) of the translation diagram of the BMS-learner \(M\) to it. In \(V(\sigma)\), which is being built after having observed \(\sigma\), we keep track of all states visited so far together with the datum which caused the first transfer to the respective state. In order to assure \(\text{Ex}\)-convergence, we do not change the subgraph in case the new state had already been visited after some proper initial segment of \(\sigma\) was observed. From \(V(\sigma)\) we can reconstruct the last first-time-visited state \(s^*_M(\sigma)\) of \(M\) while observing the equivalent sequence corresponding to \(\sigma\). Moreover, we build the equivalent text \(\hat{T}\) by inserting a path of already observed data leading to state \(s^*_M(\sigma)\), in case this is necessary to prevent the learner \(M^\text{It}\) from returning to a previously visited state but the last one. With this strategy we make sure that the last state is the one we are actually in, as keeping track of the current state while observing the original text may destroy the \(\text{Ex}\)-convergence.

Formally, we define functions \(\text{pump} : \Sigma^{<\omega}\{\varepsilon\} \times \mathbb{N} \to \Sigma^{<\omega}\) and \(V : \Sigma^{<\omega} \to \Sigma^{<\omega}\) by

\[
\begin{align*}
\text{pump}(V(\sigma), x) & = \begin{cases} 
\varepsilon, & \text{if } s_M(s^*_M(\sigma), x) \notin \text{pr}_1[V(\sigma)]; \\
x \cdot \text{path}(s_M(s^*_M(\sigma), x), s^*_M(\sigma)), & \text{otherwise};
\end{cases} \\
V(\varepsilon) & = \varepsilon;
\end{align*}
\]
Let \(\beta\) learn in the proof of Theorem 3.1. Again for \(s_0, s_1 \in S\) with \(\text{path}(s_0, s_1)\) we refer to the unique sequence \((\sigma(i), \sigma(i+1), \ldots, \sigma(j))\) of second coordinates in \(V(\sigma)\) such that \((s_0, \sigma(i)) \cdots \hat{\sigma} \sigma(j))\) is an intermediate sequence in \(V(\sigma)\). The learner \(M_{\text{It}}\) is now defined by

\[
M_{\text{It}}(\sigma^\ast x) = \text{pad}(h_{M}^\ast(s_{M_{\text{It}}}^\ast(\sigma), \text{pump}(V(\sigma)), x), V(\sigma^\ast x)).
\]

By construction \(s_{M_{\text{It}}}^\ast(\sigma) = \text{last}(\text{pr}_1(V(\sigma)))\) and therefore the hypothesis of \(M_{\text{It}}\) on some sequence \(\sigma^\ast x\) is always only based on \(V(\sigma)\) and \(x\), which makes \(M_{\text{It}}\) iterative.

The text \(\hat{T} = \bigcup_{t \in \mathbb{N}} \tau_t\) with \(\tau_0 = \epsilon\) and \(\tau_{i+1} = \tau_i \text{pump}(V(T[t]), T(t))\) is a text for \(L\). Let \(s : \mathbb{N} \to \mathbb{N}, t \mapsto |\tau_t|\) be the corresponding simulating function. As for all \(t \in \mathbb{N}\) holds content\((T[t]) = \text{content}(\hat{T}[s(t)])\) and \(M_{\text{It}}(T[t]) = \text{pad}(M(T[s(t)]), V(T[t]))\), we obtain \(W_{M_{\text{It}}}(T[t]) = W_{M}(T[s(t)])\) and because \(\delta\) is semantic and afoot, we conclude the semantic \(\delta\)-convergence of \(M_{\text{It}}\) on \(T\). Having in mind that \(M\) uses only finitely many pairwise distinct states \(V(T[t])\) stabilizes. Paired with the \(\text{Ex}\)-convergence of \(M\) on \(\hat{T}\) we conclude the \(\text{Ex}\)-convergence of \(M_{\text{It}}\) on \(T\).

Note that obviously the proof is identical for learning from positive and negative information, introduced by [Gold67]. In this learning model the information the learner receives is labeled, like in binary classification, always only based on \(V(\sigma)\) and \(x\), which makes \(M_{\text{It}}\) iterative.

With Lemma 3.1 the following results transfer from learning with iterative learners and it remains to investigate the relations to and between the non-semantic requirements Conv, SDec and SNU.

**Theorem 3.2.**

1. \([\text{TxtBMS}_s \text{NUEx}] = [\text{TxtBMS}_s \text{Ex}]\)
2. \([\text{TxtBMS}_s \text{DecEx}] = [\text{TxtBMS}_s \text{WMonEx}] = [\text{TxtBMS}_s \text{CautEx}] = [\text{TxtBMS}_s \text{Ex}]\)
3. \([\text{TxtBMS}_s \text{MonEx}] \subset [\text{TxtBMS}_s \text{Ex}]\)
4. \([\text{TxtBMS}_s \text{SMonEx}] \subset [\text{TxtBMS}_s \text{MonEx}]\)

**Proof.** The respective results for iterative learners can be found in [CM08 Theorem 2], [JKMS16 Theorem 10], [JKMS16 Theorem 3] and [JKMS16 Theorem 2].

---

4. **RELATIONS TO AND BETWEEN SYNTACTIC LEARNING REQUIREMENTS**

The following lemma establishes that we may assume \(\text{BMS}_s\)-learners to never go back to withdrawn states. This is essential in almost all of the following proofs. It can also be used to simplify the proof of Lemma 3.1.

**Lemma 4.1.** Let \(\beta\) be a learning success criterion allowing for simulation on equivalent text and \(L \in [\text{TxtBMS}_s \beta]\). Then there is a \(\text{BMS}\)-learner \(N\) such that \(N\) never returns to a withdrawn state and \(\text{BMS}_s\beta\)-learns \(L\) from texts.

**Proof.** Let \(M\) be a \(\text{BMS}\)-learner with \(L \in \text{TxtBMS}_s \beta(M)\). We employ a construction similar to the one in the proof of Theorem 3.1. Again for \(V \in (\Sigma^\omega Q \times \Sigma)\) with pairwise distinct first coordinates and \(s' \in \text{pr}_1[V]\) by \(\text{path}(V, s')\) we denote the unique sequence of second coordinates \(x_0^\ast \cdots \cdots \hat{x}_t\) of \(V\) such that \((s', x_0^\ast \cdots \cdots \hat{x}_t)\) is a final segment of \(V\). The \(\text{BMS}\) learner \(N\) is initialized with state \(\text{pad}(0, (0, \#))\) and for every \(s \in Q, V \in (\Sigma^\omega Q \times \Sigma)\) and \(x \in \Sigma\) defined by

\[
s_N(s, V, x) = \begin{cases} 
(s, V), & \text{if } s_M(s, x) \in \text{pr}_1[V] \\
(s_M(s, x), V \cdots \hat{s}_M(s, s, x, x)), & \text{otherwise}
\end{cases}
\]
\[ h_N(\langle s, V \rangle, x) = \begin{cases} 
  h^*_M(s, x \cdot \text{path}(V, s_M(s, x))), & \text{if } s_M(s, x) \in \text{pr}_1[V]; 
  h_M(s, x), & \text{otherwise.}
\end{cases} \]

By construction \( N \) is a BMS\(_a\)-learner, as it only uses states \( \langle s, V \rangle \) where \( s = \text{pr}_1(\text{last}(V)) \) is a state used by \( M \) and for every \( s \in Q \), visited by \( M \), there is exactly one sequence \( V \in (\omega Q \times \Sigma) \) such that \( \langle s, V \rangle \) is used by \( N \). The learner \( N \) simulates \( M \) on an equivalent text just as in the proof of Theorem 3.1.

We show that strongly monotonically BMS\(_a\)-learnability does not imply strongly non-U-shapedly BMS\(_a\)-learnability.

**Theorem 4.2.** [TxtBMS, SMonEx] \( \Leftrightarrow \) [TxtBMS, SNUEx]

**Proof.** We define a self-learning BMS\(_a\)-learner \( M \) and with a tailored ORT-argument there can not be a BMS\(_a\)-learner strongly non-U-shapedly learning all languages that \( M \) learns strongly monotonically.

Consider the BMS\(_a\)-learner \( M \) initialized with state \( \langle ?, \langle \emptyset \rangle \rangle \) and \( h_M \) and \( s_M \) for every \( e \in \Omega, D \subseteq \mathbb{N} \) finite and \( x \in \Sigma \) defined by:

\[
\begin{align*}
  s_M(e, D), x) &= \begin{cases} 
    \langle e, \langle D \rangle \rangle, & \text{if } x \in D \cup \{\#\} \lor \varphi_x(e) = e; \\
    \langle \varphi_x(e), D \cup \{x\} \rangle, & \text{else if } \varphi_x(e) \neq e; \\
    \uparrow, & \text{otherwise.}
  \end{cases}
\end{align*}
\]

\[
\begin{align*}
  h_M(e, D), x) &= \begin{cases} 
    e, & \text{if } x \in D \cup \{\#\} \lor \varphi_x(e) = e; \\
    \varphi_x(e), & \text{else if } \varphi_x(e) \neq e; \\
    \uparrow, & \text{otherwise.}
  \end{cases}
\end{align*}
\]

Thus, \( M \) is self-learning by interpreting the datum \( x \) as a program and the conjectures are generated by applying this program to the last hypothesis. (We identify \( \varphi_x \) with the function obtained by using a bijection from \( \mathbb{N} \) to \( \Omega \).) Further, in form of the states, the last hypothesis as well as exactly the data that already lead to a mind change of \( M \) is stored.

Let \( \mathcal{L} = \text{TxtBMS\(_a\), SMonEx}(M) \).

Assume there is a BMS\(_a\)-learner \( N \) with hypothesis generating function \( h_N \) and state transition function \( s_N \), such that \( \mathcal{L} \subseteq \text{TxtBMS\(_a\), SNUEx}(N) \). By Lemma 3.1 we assume that \( N \) does not return to withdrawn states.

We are going to obtain a language \( L \in \mathcal{L} \) not strongly non-U-shapedly learned by \( N \) by applying 1-1 ORT and thereby referring to the c.e. predicates MC and NoMC defined for fixed \( a, b \in \mathcal{R} \), all \( k \in \mathbb{N} \) and \( \sigma \in \Sigma^\omega \) with the help of the formulas \( \psi_k(\ell) \), expressing that the BMS\(_a\)-learner \( N \) does not perform a mind- or state-change on the text \( a[k] \cdot b(k) \cdot \#^{\omega} \) after having observed \( a[k] \cdot b(k) \cdot \#^{\ell} \). The predicates state that \( N \) does converge and (not) make a mind-change when observing \( \sigma \) after having observed \( a[k] \cdot a(k) \cdot \#^{\ell} \), with \( \ell_k \) being the least \( \ell \) with \( \psi_k(\ell) \).

\[
\psi_k(\ell) \Leftrightarrow N(a[k] \cdot b(k) \cdot \#^{\ell-1}) \land s_N^*(a[k] \cdot b(k) \cdot \#^{\ell}) = s_N^*(a[k] \cdot b(k) \cdot \#^{\ell+1});
\]

\[
\text{NoMC}(k, \sigma) \Leftrightarrow \exists \ell_k \in \mathbb{N} (\psi_k(\ell_k) \land \forall \ell < \ell_k \neg \psi_k(\ell) \land N(a[k] \cdot b(k) \cdot \#^{\ell} \cdot \sigma) \downarrow \neq N(a[k] \cdot b(k) \cdot \#^{\ell_k});
\]

\[
\text{MC}(k, \sigma) \Leftrightarrow \exists \ell_k \in \mathbb{N} (\psi_k(\ell_k) \land \forall \ell < \ell_k \neg \psi_k(\ell) \land N(a[k] \cdot b(k) \cdot \#^{\ell} \cdot \sigma) \downarrow \neq N(a[k] \cdot b(k) \cdot \#^{\ell_k}).
\]

Now, let \( p \) be an index for the program which on inputs \( k \in \mathbb{N} \) and \( \sigma \in \Sigma^\omega \) searches for \( \ell_k \). In case \( \ell_k \) exists, the program encoded in \( p \) runs on \( a[k] \cdot b(k) \cdot \#^{\ell_k} \cdot \sigma \). Hence, \( \Phi_p(k, \sigma) \) stands for the number of computation steps the program just described needs on input \( k, \sigma \). By the definition of \( p \) we have \( \Phi_p(k, \sigma) \uparrow \) if and only if \( \ell_k \uparrow \) or \( N(a[k] \cdot b(k) \cdot \#{\ell_k} \cdot \sigma) \uparrow \).

We abbreviate with \( (\omega a, i) = \omega \cdot a \cup \{\#\} \) the set of all finite sequences over \( \omega \cdot a \cup \{\#\} \) with length at most \( i \). Moreover, we employ a well-order \( <_a \) on \( (\omega \cdot \text{ran}(a)) \) by letting \( p <_a \sigma \) if and only if for the unique \( i_p \) such that \( p \in (\omega a, i_p + 1) \) holds \( \sigma \notin (\omega a, i_p + 1) \) or else \( \sigma \notin (\omega a, i_p) \) and at the same time \( \langle p \rangle < \langle \sigma \rangle \). For constructing \( L \) we will also make use of the c.e. sets

\[
E_k = \{a(i) \mid \forall \sigma < (\omega a, i) \text{ NoMC}(k, \sigma) \lor (\exists \sigma < p <_a \sigma \text{ NoMC}(k, p) \land \Phi_p(k, \sigma) > i)} \}.
\]
It is easy to see that $E_k$ is finite and equals $\{a(i) \mid i < \max(\{i_{\sigma_0} \cup \{\Phi_p(k, \sigma) \mid \sigma \leq_a \sigma_0\})\}$ if and only if for
$
\sigma_0 \in (\omega^\omega \text{ran}(a))$ holds MC($k, \sigma_0$) and NoMC($k, \sigma$) for all $\sigma \leq_a \sigma_0$. Otherwise $E_k = \text{ran}(a)$.

By 1-1 ORT there are $a, b, c_1, c_2 \in \mathbb{R}$ with pairwise disjoint ranges and $e_0 \in \mathbb{N}$, such that

$$\varphi_{a(i)}(e) = \begin{cases} e_0 & \text{if } e \in \{?, e_0\}; \\ e_2(k) & \text{else if } e = e_1(k) \text{ for some } k \leq i; \\ e, & \text{otherwise}; \end{cases}$$

$$\varphi_{b(k)}(e) = \begin{cases} e_1(k) & \text{if } e \in \{?, e_0\}; \\ e, & \text{otherwise}; \end{cases}$$

$$W_{e_0} = \begin{cases} \text{ran}(a[t_0]) & \text{if } t_0 \text{ is minimal with } \forall t \geq t_0 \ (N(a[t]) = N(a[t_0]) \wedge s_N(a[t]) = s_N(a[t_0])); \\ \text{ran}(a), & \text{no such } t_0 \text{ exists};. \end{cases}$$

$$W_{e_1(k)} = \text{content}(a[k]) \cup \{b(k)\} \cup \begin{cases} E_k & \text{if } \exists \sigma_0 \ (\text{MC}(k, \sigma_0) \wedge \forall \sigma \leq_a \sigma_0 \ \text{NoMC}(k, \sigma)); \\ \emptyset, & \text{otherwise}; \end{cases}$$

$$W_{e_2(k)} = \text{content}(a[k]) \cup \{b(k)\} \cup E_k.$$ As $W_{e_0} \in \mathcal{L}$ by construction, $N$ has to learn it and hence $t_0$ exists.

We first observe that there exists $\sigma_0$ such that MC($t_0, \sigma_0$) and NoMC($t_0, \sigma$) for all $\sigma \leq_a \sigma_0$. Assume otherwise, then either $t_0$ or for all $\sigma \in (\omega^\omega \text{ran}(a))$ holds NoMC($t_0, \sigma$) or for $\sigma_0$ minimal with $\neg$NoMC($t_0, \sigma_0$) we have $N(a[t_0]) \wedge b(t_0) \neg \#^{t_0 \sigma}$. Anyhow, this would mean $E_{t_0} = \text{ran}(a)$. By the definition of $e_1, e_2$ and our converse assumption we obtain $W_{e_1(t_0)} = \text{content}(a[t_0]) \cup \{b(t_0)\}$ and $W_{e_2(t_0)} = \text{ran}(a) \cup \{b(t_0)\}$. It can be easily checked that $W_{e_1(t_0)}$ and $W_{e_2(t_0)}$ are strongly monotonically learned by $M$ and hence lie in $\mathcal{L}$. As $N$ has to learn $W_{e_1(t_0)}$ from the text $a[t_0] \wedge b(t_0) \neg \#^{t_0 \sigma}$, we know $t_0$ and moreover $W_N(a[t_0] \wedge b(t_0) \neg \#^t) = W_{e_1(t_0)}$ holds for all $t \geq t_0$. Moreover, $N$ has to learn $W_{e_2(t_0)}$ from all the texts $a[t_0] \wedge b(t_0) \neg \#^{t_0 \sigma}$ with $\sigma \in (\omega^\omega \text{ran}(a))$. Thus, $N(a[t_0] \wedge b(t_0) \neg \#^{t_0 \sigma}) \downarrow$ for all $\sigma \in (\omega^\omega \text{ran}(a))$. Because of our converse assumption, the only option left is NoMC($t_0, \sigma$) for all $\sigma \in (\omega^\omega \text{ran}(a))$. Since this is equivalent to $N(a[t_0] \wedge b(t_0) \neg \#^{t_0 \sigma}) = N(a[t_0] \wedge b(t_0) \neg \#^{t_0 \sigma})$ for all $\sigma \in (\omega^\omega \text{ran}(a))$, $N$ cannot learn both $W_{e_1(t_0)}$ and $W_{e_2(t_0)}$. Hence $\sigma_0$ exists.

By the choice of $t_0$ and $\sigma_0$ we obtain $E_{t_0} = \text{content}(a[t_1])$ for $t_1 = \max(\{t_{\sigma_0} \cup \{\Phi_p(k, \sigma) \mid \sigma \leq_a \sigma_0\}) \in \mathbb{N}$. Let $\hat{t} = \max\{t_0, t_1\}$ and $L = \text{content}(a[\hat{t}]) \cup \{b(t_0)\}$. Then $W_{e_1(t_0)} = W_{e_2(t_0)} = L \in \mathcal{L}$ and by construction of $E_{t_0}$ we have Cons($\sigma_0, L$). Because of $\hat{t} \geq t_0$, we obtain $s_N^\sigma(a[\hat{t}]) = s_N^\sigma(a[t_0])$. With this and the choice of $t_0$ we conclude $N(a[\hat{t}] \wedge b(t_0) \neg \#^t) = N(a[t_0] \wedge b(t_0) \neg \#^t)$ for all $t \in \mathbb{N}$. Further, as $N$ learns $L$ from the text $a[\hat{t}] \wedge b(t_0) \neg \#^\omega$, we have $W_N(a[\hat{t}] \wedge b(t_0) \neg \#^\omega) = L$. On the other hand by MC($t_0, \sigma_0$) we obtain $N(a[\hat{t}] \wedge b(t_0) \neg \#^t) \neq N(a[\hat{t}] \wedge b(t_0) \neg \#^t \sigma_0)$, which forces $N$ to perform a syntactic U-shape on the text $a[\hat{t}] \wedge b(t_0) \neg \#^t \sigma_0 \neg \omega$ for $L$.

For inferring the relations between the syntactic learning requirements SNU, SDec and Conv, we refer to [WB]. All these criteria are closely related to strongly locking learners, which we define in the following.

It was observed by [BB78] that the learnability of every language $L$ by a learner $M$ is witnessed by a sequence $\sigma$, consistent with $L$, such that $M(\sigma)$ is an index for $L$ and no extension of $\sigma$ consistent with $L$ will lead to a mind-change of $M$. Such a sequence $\sigma$ is called (sink-)locking sequence for $M$ on $L$. For a similar purpose as ours [JKMS10] introduced strongly locking learners. A learner $M$ acts strongly locking on a language $L$, if for every text $T$ for $L$ there is an initial segment $\sigma$ of $T$ that is a locking sequence for $M$ on $L$.

The proof of the following theorem generalizes the construction of a conservative and strongly decisive iterative learner from a strongly locking iterative learner in [JKMS10] Theorem 8]. With it we obtain in the Corollary thereafter, that all non-semantic learning restrictions coincide.

**Theorem 4.3.** Let $\mathcal{L}$ be a set of languages BMS$_e$Ex-learned by a strongly locking BMS-learner. Then

$\mathcal{L} \in [\text{TxtBMS}_e \text{WbEx}].$
Proof. Let $\mathcal{L} \in [\text{TxtBMS}_{s}\text{Ex}]$ be learned by the strongly locking learner $M$. By Lemma 4.1 we may assume that $M$ does not return to withdrawn states.

We proceed in two steps. First we construct a learner $M'$ conservatively $\text{BMS}_{s}\text{Ex}$-learning at least $\mathcal{L}$ in a strong sense, i.e.,

$$\forall \sigma \in \Sigma^{<\omega} \forall x \in \Sigma \left( M'(\sigma^{-} x) \neq M'(\sigma) \Rightarrow x \not\in W_{M'(\sigma)} \right).$$

That we require the last datum to violate consistency with the former hypothesis fits the setting of $\text{BMS}$-learners and is also called locally conservative by [JLZ06]. Second, with such a learner at hand, we are going to construct a learner $N$ which $\text{BMS}_{s}\text{Ex}$-learns $\mathcal{L}$ in a witness-based fashion. We will do this by keeping track of all data having caused a mind-change so far. More concretely, we alter the text by excluding mind-change data causing another mind-change and make sure that the witness for the mind-change is contained in all future hypotheses.

For defining the strongly conservative learner $M'$, we employ a one-one function $f : \mathbb{N} \times Q \to \Omega$ satisfying

$$W_{f(e,s)} = \bigcup_{t \in \mathbb{N}} W_{e}^{t}, \text{ if } \forall x \in W_{e}^{t}(h_{M}(s,x) = e \land s_{M}(s,x) = s);$$

for every hypothesis $e \in \mathbb{N} \subseteq \Omega$ and state $s \in Q$. The existence of $f$ is granted by the smn theorem. Thus, $f$ takes into account only the initial part of $W_{e}$ not necessary to possibly justify a mind-change or state-change later on. Now define for all $\sigma \in \Sigma^{<\omega}$

$$M'(\sigma) = f(M(\sigma), s_{M}(\sigma)).$$

As $M$ never returns to withdrawn states and behaves strongly locking while $\text{BMS}_{s}\text{Ex}$-learning $\mathcal{L}$, $M'$ also $\text{Ex}$-learns $\mathcal{L}$. For $\sigma \neq e$ the values of $M(\sigma)$ and $s_{M}(\sigma)$ only depend on $s_{M}(\sigma^{-})$ and last($\sigma$) and hence $M'$ is a $\text{BMS}_{s}$-learner with $s_{M} = s_{M}$. Moreover, by construction it is conservative in the strong sense defined in (1).

We now define the witness-based learner $N$. In addition to thinning out the hypotheses of $M'$, as we did with the hypotheses of $M$ when constructing $M'$ from $M$, we patch all data causing mind-changes to it. This data is stored in the states used by $N$. Further, we only alter our old hypothesis in case we can guarantee the existence of a witness justifying the possible mind-change. To do this in a computable way, we need to store also the last hypothesis of $M'$ in the states of $N$.

For every datum $x \in \Sigma$, data-sequence $\sigma \in \Sigma^{<\omega}$, hypothesis $e \in \mathbb{N} \subseteq \Omega$ and every finite sequence $MC$ of natural numbers, interpreted as pairs of hypotheses and data, we define a state transition function $s_{N}$, auxiliary hypothesis generating function $h_{M}$, recursive function $g : \mathbb{N}^{2} \to \Omega$ and the learner $N$ by

$$h(\langle s, \langle \text{MC} \rangle \rangle, x) = \begin{cases} h_{M}(s, \#), & \text{if } x \in \text{pr}_{2}[\text{MC}]; \\ h_{M}(s, x), & \text{otherwise}; \end{cases}$$

$$s_{N}(\langle s, \langle \text{MC} \rangle \rangle, x) = \begin{cases} \langle s_{M}(s, \#), \langle \text{MC} \rangle \rangle, & \text{if } x \in \text{pr}_{2}[\text{MC}] \land h_{M}(s, \#) = \text{pr}_{1}(\text{last}(\text{MC})); \\ \langle s_{M}(s, x), \langle \text{MC} \rangle \rangle, & \text{if } x \in \text{pr}_{2}[\text{MC}] \land h_{M}(s, \#) \neq \text{pr}_{1}(\text{last}(\text{MC})); \\ \langle s_{M}(s, x), \langle \text{MC} \rangle \rangle, & \text{else if } h_{M}(s, x) = \text{pr}_{1}(\text{last}(\text{MC})); \\ \langle s_{M}(s, x), \langle \text{MC} \rangle \rangle, & \text{otherwise}; \end{cases}$$

$$W_{g(e, \langle s, \langle \text{MC} \rangle \rangle)} = \text{pr}_{2}[\text{MC}] \cup W_{e};$$

$$N(\sigma^{-} x) = \begin{cases} ?, & \text{if } h^{*}(\sigma^{-} x) = ?; \\ g(h^{*}(\sigma^{-} x), s_{N}^{*}(\sigma^{-} x)), & \text{else if } h^{*}(\sigma^{-} x) \neq \text{pr}_{1}(\text{last}(\text{decode}(\text{pr}_{2}(s_{N}^{*}(\sigma)))))); \\ N(\sigma), & \text{otherwise}. \end{cases}$$

Thus with the help of $g$ the data stored in the second coordinates of $\text{MC}$ is patched to the language encoded in $e$. Further, $N$ only makes a mind-change if $h^{*}$ does, as $h^{*}(\sigma) = \text{pr}_{1}(\text{last}(\text{decode}(\text{pr}_{2}(s_{N}^{*}(\sigma))))))$. The learner $h^{*}$ behaves like $M'$ on the text, in which every datum repeatedly causing a mind-change is replaced by the pause symbol.
Let \( L \in \mathcal{L} \) and \( T \in \text{Txt}(L) \). It is easy to see that for the text \( T' \) recursively defined by

\[
T'(t) = \begin{cases} 
\#, & \text{if } \exists s < t \ (T(s) = T(t) \land M'(T'[s] \Delta T(s)) \neq M'(T'[s])); \\
T(t), & \text{otherwise,}
\end{cases}
\]

holds \( h^*(T[t]) = M'(T'[t]) \) for all \( t \in \mathbb{N} \). This follows with a simultaneous induction also showing \( \text{pr}_1(s_N^*(T[t])) = s_{M'}^*(T'[t]) \). Hence \( h^* \) on \( T \) behaves like \( M' \) on \( T' \in \text{Txt}(L) \).

Because \( M' \) converges on \( T' \), it makes only finitely many mind-changes and uses only finitely many states, which implies that \( N \) also only uses finitely many states. Let \( e = M'(T'[t_0]) \) be the final correct hypothesis of \( M' \) on \( T' \) with \( t_0 \in \mathbb{N} \) chosen appropriately. Because \( M' \) never returns to withdrawn states, the states of \( N \) also stabilize. Moreover, \( N(T[t_0]) \) has to be correct since \( \text{pr}_2[MC] \subseteq W_2 \).

As already mentioned, \( N \) learns every \( L \in \mathcal{L} \) witness-based because \( M' \) is strongly conservative. Every time \( N \) performs a mind-change on \( T \), so does \( M' \) on \( T' \). Therefore, there is a responsible datum \( x \) which was not in the former hypothesis of \( M' \) and also has not occurred so far, as no datum in \( T' \) causes more than one mind-change. This datum \( x \) will be contained in all languages hypothesized by \( N \) in the future. \( \square \)

With the latter theorem it is straightforward to observe that in the \( \text{BMS}_s \text{Ex} \)-setting conservative, strongly decisive and strongly non-U-shaped \( \text{Ex} \)-learning are equivalent.

**Corollary 4.4.** We have \([\text{TxtBMS}_s \text{ConvEx}] = [\text{TxtBMS}_s \text{SDecEx}] = [\text{TxtBMS}_s \text{SNUEx}]\).

**Proof.** On the one hand a conservative or strongly decisive learning behavior is also a strongly non-U-shaped learning behavior. On the other hand, a learner behaving strongly non-U-shaped proceeds strongly locking and, by Theorem 4.3 from a strongly locking learner we may construct a learner with at least equal learning power, acting witness-based and hence also conservatively and strongly decisively. \( \square \)

By [JKMS16] Theorem 2] and Lemma 3.1] (1) we obtain

\[ [\text{TxtBMS}_s \text{ConvEx}] \not\subseteq [\text{TxtBMS}_s \text{SMonEx}]. \]

From this we conclude with Theorem 4.2 and Corollary 4.4 the following incomparability

\[ [\text{TxtBMS}_s \text{ConvEx}] \perp [\text{TxtBMS}_s \text{SMonEx}]. \]

Similarly, with [JKMS16] Theorem 3] and again Lemma 3.1] (1) we obtain \([\text{TxtBMS}_s \text{ConvEx}] \not\subseteq [\text{TxtBMS}_s \text{MonEx}]\). As Theorem 4.2 implies \([\text{TxtBMS}_s \text{MonEx}] \not\subseteq [\text{TxtBMS}_s \text{SNUEx}]\), with Corollary 4.4 follows

\[ [\text{TxtBMS}_s \text{ConvEx}] \perp [\text{TxtBMS}_s \text{MonEx}]. \]

Because Theorem 4.2 also reproves \([\text{TxtBMS}_s \text{SNUEx}] \subseteq [\text{TxtBMS}_s \text{Ex}]\), first observed in [CK13, Th. 3.10], we completed the map for \( \text{BMS}_s \text{Ex} \)-learning from texts. An overview is depicted in Figure 1.

As this map equals the one for \( \text{It} \)-learning, naturally the question arises, whether a result similar to Lemma 3.1] can be observed for the syntactic learning criteria. In the following we show that this is not the case.

**Theorem 4.5.** \([\text{It} \text{TxtSNUEx}] \subseteq [\text{TxtBMS}_s \text{SNUEx}]\)

**Proof.** By Lemma 3.1] we have \([\text{It} \text{TxtSNUEx}] \subseteq [\text{TxtBMS}_s \text{SNUEx}]\).

We consider the \( \text{BMS} \)-learner \( M \) initialized with state \( \langle \langle ?, 0 \rangle, \langle 0 \rangle \rangle \) and \( h_M \) and \( s_M \) for every \( \langle e, \xi \rangle \in \Omega \), \( D \subseteq \mathbb{N} \) finite and \( x \in \Sigma \) defined by:

\[
s_M(\langle \langle e, \xi \rangle, \langle D \rangle \rangle, x) = \begin{cases} 
\langle \langle e, \xi \rangle, \langle D \rangle \rangle, & \text{if } x \in D \cup \{\#\} \lor \\
\varphi_x(\langle \rho, \xi \rangle), \langle D \cup \{x\} \rangle, & \text{if } \text{pr}_1(\rho(\langle \rho, \xi \rangle) \downarrow) = e; \\
\varphi_x(\langle \rho, \xi \rangle), \langle D \cup \{x\} \rangle, & \text{else if } \text{pr}_1(\rho(\langle \rho, \xi \rangle) \downarrow) \neq e; \\
\uparrow, & \text{otherwise.}
\end{cases}
\]

![Figure 1](image-url)
Figure 1. Relations between delayable learning restrictions in explanatory finitely bounded memory states learning of languages from informants. The arrows represent implications independent of the model. The outlined areas stand for equivalence classes with respect to learning power, when the underlying model is $\text{TxtBMS}_{\text{Ex}}$.

Additionally to the last hypothesis as well as exactly the data that already lead to a mind-change of $M$, some parameter $\xi$ is stored, indicating whether a further mind-change may cause a syntactic $U$-shape.

Let $\mathcal{L} = \text{TxtBMS}_{\text{SNUEx}}(M)$. We will show that there is no iterative learner $\text{ItTxtSNUEx}$-learning $\mathcal{L}$. Assume $N$ is an iterative learner with hypothesis generating function $h_N$ and $\mathcal{L} \subseteq \text{ItTxtEx}(N)$.

We obtain $L \in \mathcal{L}\setminus\text{ItTxtSNUEx}(N)$ by applying 1-1 ORT [Cas74] referring to the $\Sigma_1$-predicates MC and NoMC, expressing that $N$ does (not) perform a mind-change on a text built from parameters $a,b \in \mathcal{R}$. More specifically, the predicates state that $N$ does converge and (not) make a mind-change when observing $\sigma \in \Sigma^{<\omega}$ after having observed $a[i] \overset{\sim}{\circ} b(i) \overset{\sim}{\circ} \#^\ell$, with $i \in \mathbb{N}$.

$$h_M(\langle e, \xi, \langle D \rangle \rangle, x) = \begin{cases} 
eq e, & \text{if } x \in D \cup \{\#\} \quad \lor \quad \text{pr}_1(\varphi_x(\langle e, \xi \rangle) \downarrow) = e; \\ \text{pr}_1(\varphi_x(\langle e, \xi \rangle) \downarrow), & \text{else if } \text{pr}_1(\varphi_x(\langle e, \xi \rangle) \downarrow) \neq e; \\ \uparrow, & \text{otherwise.} \end{cases}$$
By 1-1 ORT, applied to the recursive operator implicit in the following case distinction, there are recursive total functions $a, b, e_1, e_2$ with pairwise disjoint ranges and $e_0 \in \mathbb{N}$, such that for all $i, \xi \in \mathbb{N}$, $e \in \Omega$

$$
\varphi_{a(i)}(\langle e, \xi \rangle) = \begin{cases} 
\langle e_0, \xi \rangle, & \text{if } e \in \{?, e_0\}; \\
\langle e_1(k), 1 \rangle, & \text{else if } \xi = 0, i \text{ even and } \exists k \leq i \ (e = e_1(k)); \\
\langle e_1(k), 2 \rangle, & \text{else if } \xi = 0, i \text{ odd and } \exists k \leq i \ (e = e_1(k)); \\
\langle e_2(k), 0 \rangle, & \text{else if } \xi = 1, i \text{ odd and } \exists k \leq i \ (e = e_1(k)); \\
\langle e_2(k), 0 \rangle, & \text{else if } \xi = 2, i \text{ even and } \exists k \leq i \ (e = e_1(k)); \\
(\langle e, \xi \rangle), & \text{otherwise};
\end{cases}
$$

$$
\varphi_{b(i)}(\langle e, \xi \rangle) = \begin{cases} 
\langle e_1(i), \xi \rangle, & \text{if } e \in \{?, e_0\}; \\
\langle e, \xi \rangle, & \text{otherwise};
\end{cases}
$$

$$W_{e_0} = \begin{cases} 
\text{ran}(a[t_0]), & \text{if } t_0 \text{ is minimal with } \forall t \geq t_0 \ N(a[t]) = N(a[t_0]); \\
\text{ran}(a), & \text{no such } t_0 \text{ exists};
\end{cases}
$$

$$W_{e_1(i)} = \text{ran}(a[i]) \cup \{b(i)\} \cup \begin{cases} 
\{a(j)\} & \text{for first } j \geq i \text{ found} \\
\varnothing & \text{with MC}(i, a(j));
\end{cases}
$$

$$W_{e_2(i)} = \text{ran}(a) \cup \{b(i)\}.
$$

As the learner constantly puts out $e_0$ on every text for $W_{e_0}$, we have $W_{e_0} \in \mathcal{L}$. Thus, also $N$ learns the finite language $W_{e_0}$ and $t_0$ exists. Note that by the iterativeness of $N$ we obtain $N(a[t_0]) = N(a[t_0] \upharpoonright a(i))$ for all $i \geq t_0$ and with this $N(a[t_0] \upharpoonright b(t_0) \upharpoonright \#^{t_0}) = N(a[t_0] \upharpoonright a(i) \upharpoonright b(t_0) \upharpoonright \#^{t_0})$ for all $i \geq t_0$.

$W_{e_1(t_0)}$ and $W_{e_2(t_0)}$ also lie in $\mathcal{L}$. To see that $M$ explanatory learns both of them, note that, after having observed $b(t_0)$, $M$ only changes its mind from $e_1(t_0)$ to $e_2(t_0)$ after having seen $a(i)$ and $a(j)$ with $i, j \geq t_0$ and $i \in 2\mathbb{N}$ as well as $j \in 2\mathbb{N} + 1$. This clearly happens for every text for the infinite language $W_{e_2(t_0)}$: As $|W_{e_1(t_0)} \setminus \{\text{content}(a[t_0]) \cup \{b(t_0)\}\}| \leq 1$, this mind change never occurs for any text for $W_{e_2(t_0)}$.

The syntactic non-U-shapedness of $M$’s learning processes can be easily seen as for all $k, l \in \mathbb{N}$ the languages $W_{e_0}$, $W_{e_1(k)}$ and $W_{e_2(l)}$ are pairwise distinct, the learner never returns to an abandoned hypothesis and $M$ only leaves hypothesis $\langle e_1(k), 0 \rangle$ for $\langle e_1(k), \xi \rangle$, $\xi \neq 0$, if $W_{e_1(k)}$ is not correct.

Next, we show the existence of $j \geq t_0$ with $\text{MC}(t_0, a(j))$. Assume towards a contradiction that $j$ does not exist. Then $W_{e_1(t_0)} = \text{content}(a[t_0]) \cup \{b(t_0)\}$. As $M$ learns this language from the text $a[t_0] \upharpoonright b(t_0) \upharpoonright \#^{\infty}$, so does $N$. The convergence of $N$ implies the existence of $t_0$. Thus, for every $j \in \mathbb{N}$ we either have $N(a[t_0] \upharpoonright b(t_0) \upharpoonright \#^{t_0} \upharpoonright a(j)) = N(a[t_0] \upharpoonright b(t_0) \upharpoonright \#^{t_0})$ or the computation of $N(a[t_0] \upharpoonright b(t_0) \upharpoonright \#^{t_0} \upharpoonright a(j))$ does not terminate. Because $N$ is iterative and learns $W_{e_2(t_0)}$, it may not be undefined and therefore always the latter is the case. But then $N$ will not learn $W_{e_1(t_0)}$ and $W_{e_2(t_0)}$ as they are different but $N$ does not make a mind-change on the text $a[t_0] \upharpoonright b(t_0) \upharpoonright \#^{t_0} \upharpoonright a(j)$ after having observed the initial segment $a[t_0] \upharpoonright b(t_0) \upharpoonright \#^{t_0}$, due to its iterativeness. Hence, $x$ exists and $W_{e_1(t_0)} = \text{ran}(a[t_0]) \cup \{b(t_0), a(j)\}$.

Finally, by the choice of $j$, the learner $N$ does perform a syntactic U-shape on the text $a[t_0] \upharpoonright a(j) \upharpoonright b(t_0) \upharpoonright \#^{t_0} \upharpoonright a(j)$ for $W_{e_1(t_0)}$. More precisely, $t_0$ and $t_0$ were chosen such that $N(a[t_0] \upharpoonright a(j) \upharpoonright b(t_0) \upharpoonright \#^{t_0})$ has to be correct and the characterizing property of $j$ assures

$$N(a[t_0] \upharpoonright a(j) \upharpoonright b(t_0) \upharpoonright \#^{t_0}) \neq N(a[t_0] \upharpoonright a(j) \upharpoonright b(t_0) \upharpoonright \#^{t_0} \upharpoonright a(j)).$$

Thus, no iterative learner can explanatory syntactically non-U-shapedly learn the language $\mathcal{L}$.

By Corollary 6.8, we also obtain $[\text{ItTxtSDecEx}] \subseteq [\text{TxtBMS}_{\text{sSDecEx}}]$ and $[\text{ItTxtConvEx}] \subseteq [\text{TxtBMS}_{\text{ConvEx}}]$. 

13
5. Related Open Problems

We have given a complete map for learning with bounded memory states, where, on the way to success, the learner must use only finitely many states. Future work can address the complete maps for learning with an a priori bounded number of memory states, which needs very different combinatorial arguments. Results in this regard can be found in [CCJS07] and [CK13]. We expect to see trade-offs, for example allowing for more states may make it possible to add various learning restrictions (just as non-deterministic finite automata can be made deterministic at the cost of an exponential state explosion).

Also memory-restricted learning from positive and negative data (so-called informant) has only partially been investigated for iterative learners and to our knowledge not at all for other models of memory-restricted learning. Very interesting also in regard of 1-1 hypothesis spaces that prevent coding tricks is the Bem-hierarchy, see [FJO94], [LZ96] and [CJLZ99].

In the spirit of grammatical inference, we encourage to investigate the learnability of carefully chosen indexable families arising from applied machine learning or cognitive science research.

Acknowledgements. This work was supported by DFG Grant Number KO 4635/1-1. We are grateful to the people supporting us.
[Köt09] T. Kötzing. Abstraction and Complexity in Computational Learning in the Limit. PhD thesis, University of Delaware, 2009.

[KP16] T. Kötzing and R. Palenta. A map of update constraints in inductive inference. *Theoretical Computer Science*, 650:4–24, 2016.

[KS16] T. Kötzing and M. Schirneck. Towards an atlas of computational learning theory. In 33rd *Symposium on Theoretical Aspects of Computer Science*, 2016.

[KSS17] T. Kötzing, M. Schirneck, and K. Seidel. Normal forms in semantic language identification. In *Proc. of Algorithmic Learning Theory*, pages 493–516. PMLR, 2017.

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

[MPU+92] G. Marcus, S. Pinker, M. Ullman, M. Hollander, T.J. Rosen, and F. Xu. Overregularization in Language Acquisition. Monographs of the Society for Research in Child Development, vol. 57, no. 4. University of Chicago Press, 1992. Includes commentary by H. Clahsen.

[Odi99] P. Odifreddi. *Classical Recursion Theory*, volume II. Elsevier, Amsterdam, 1999.

[OSW82] D. Osherson, M. Stob, and S. Weinstein. Learning strategies. *Information and Control*, 53:32–51, 1982.

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

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

[SS82] S. Strauss and R. Stavy, editors. *U-Shaped Behavioral Growth*. Developmental Psychology Series. Academic Press, NY, 1982.

[Wie91] R. Wiehagen. A thesis in inductive inference. In *Nonmonotonic and Inductive Logic, 1st International Workshop*, Proc., pages 184–207, 1991.