Mapping Monotonic Restrictions in Inductive Inference*

Vanja Doskoč and Timo Kötzing

Hasso Plattner Institute, University of Potsdam, Germany
{vanja.doskoc, timo.koetzing}@hpi.de

Abstract. In *inductive inference* we investigate computable devices (learners) learning formal languages. In this work, we focus on *monotonic* learners which, despite their natural motivation, exhibit peculiar behaviour. A recent study analysed the learning capabilities of *strongly monotone* learners in various settings. The therein unveiled differences between *explanatory* (syntactically converging) and *behaviourally correct* (semantically converging) such learners motivate our studies of *monotone* learners in the same settings.
While the structure of the pairwise relations for monotone explanatory learning is similar to the strongly monotone case (and for similar reasons), for behaviourally correct learning a very different picture emerges. In the latter setup, we provide a *self-learning* class of languages showing that monotone learners, as opposed to their strongly monotone counterpart, do heavily rely on the order in which the information is given, an unusual result for behaviourally correct learners.

1 Introduction

Algorithmically learning a formal language from a growing but finite amount of its positive information is referred to as *inductive inference* or *language learning in the limit*. For example, a learner h (a computable device) might be presented more and more data from a formal language (a computably enumerable subset of the natural numbers), say, the set of all odd prime numbers $\mathbb{P}_o$. With each new element presented, h outputs a description for a formal language as its guess. As such, the learner may decide to conjecture a code for the set of all odd numbers $\mathbb{N}_o$. With more data given, the learner may infer some structure and finally decide to output a program for the set $\mathbb{P}_o$. If h does not change its mind any more, we say that h learned the language $\mathbb{P}_o$ correctly.

Originally introduced by Gold [9], such learning is referred to as *explanatory learning*, as the learner eventually provides a syntactically fixed explanation of the language. We denote such learning by $\mathbf{TxtGEx}$, where $\mathbf{Txt}$ indicates that the information is given from text, $\mathbf{G}$ stands for *Gold-style* or *full-information* learning and, lastly, $\mathbf{Ex}$ refers to explanatory learning. Since a single language can be learned by a learner which always guesses one and the same code for this language, we study classes of languages which can be $\mathbf{TxtGEx}$-learned by a single learner and denote the set of all such classes with $[\mathbf{TxtGEx}]$. We refer to this set as the *learning power* of $\mathbf{TxtGEx}$-learners.

Picking up the initial example, we observe that the learner h outputs a code for $\mathbb{N}_o$ overgeneralizing the target language $\mathbb{P}_o$ before outputting a correct code. The question

* This work was supported by DFG Grant Number KO 4635/1-1.

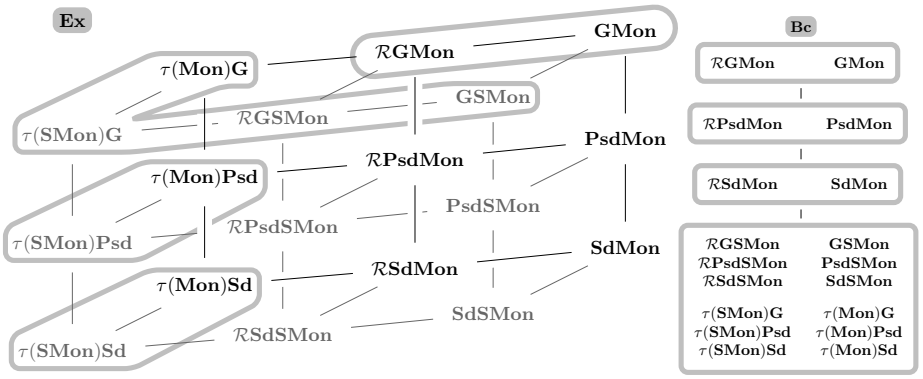
arises whether such overgeneralizations are necessary in order to obtain full learning power? Various restrictions mimicking overgeneralizations have been investigated in the literature and show such a behaviour to be crucial. A prominent example are *monotonic* learners [11,23], where the hypotheses must show a monotone behaviour. In the strongest from, the hypotheses of *strongly monotone* ($\mathbf{SMon}$) learners must form ascending chains. In a less restrictive form, only the *correctly* inferred elements, that is, elements that belong to the target language, in the hypotheses of *monotone* learners ($\mathbf{Mon}$) need to form ascending chains.

A recent study of strongly monotone learners under various additional restrictions provided a full overview of the pairwise relations between these [13]. The studied restrictions affect the data given to the learners as well as the learners themselves. In particular, the learners may be given solely the set of elements to infer their hypotheses from, referred to as *set-driven* ($\mathbf{Sd}$, [22]) learning, or may additionally be given an iteration-counter, called *partially set-driven* or *rearrangement-independent* ($\mathbf{Psd}$, [2,21]) learning. When learning indexed families of recursive languages [1] rather than classes of recursively enumerable languages, monotonic learners have been studied under similar restrictions [16,17,18]. Directly affecting the learner are requirements such as them being *total* (denoted using the prefix $\mathcal{R}$) or them being monotone on arbitrary information (denoted by the prefix $\tau(\mathbf{Mon})$).

Comparing all the possible pairwise combinations, Kötzing and Schirneck [13] show that Gold-style strongly monotone learners may be assumed so on *arbitrary* information. Besides that, they provide self-learning classes of languages [4] to show that all other combinations separate from each other. Contrasting this are their findings when studying *behaviourally correct* learners ($\mathbf{Bc}$, [5,19]), which need to provide a *semantic* explanation (rather than a syntactic one) in the limit. Behaviourally correct strongly monotone learners turn out to be equally powerful, regardless the considered restriction on the given data (that is, whether the learner has full information, is partially set-driven or set-driven) or learner itself (that is, whether it is partial, total or required to be strongly monotone on arbitrary input).

These interesting findings motivate the present study. In Section 3.1, we study monotonic explanatory learners. In particular, we observe that the overall behaviour of monotone learners resembles the one of strongly monotone learners. This similarity culminates in Theorem 3, where we prove learners which are monotone on arbitrary input, so called *globally* monotone learners, to be equal to globally strongly monotone ones. We additionally observe that most proof strategies used to separate the diverse strongly monotone learning paradigms [13] can be carried over to fit monotone learners. While these transitions are often non-trivial, they do indicate a deep similarity between these two restrictions. We provide all the necessary comparisons in Section 3.1 and depict the overall picture in a lucid map, see Figure 1(a). Please consider the full version [6] for the proofs.

In Section 3.2, we transfer the problem of finding the pairwise relations to behaviourally correct monotonic learners and discover an unexpected result. In Theorem 7, we provide a self-learning class of languages [4] using the Operator Recursion Theorem [3] showing that Gold-style monotone learners are strictly more powerful than their partially set-driven counterpart. This is particularly surprising as usually be-



(a) (b)

Fig. 1: Relation of various monotonic learning restrictions in the **(a)** explanatory (**Ex**) and **(b)** behaviourally correct (**Bc**) case. We omit mentioning **Txt** to favour readability. Solid lines imply trivial inclusions (bottom-to-top, left-to-right). Greyly edged areas illustrate a collapse of the enclosed learning criteria. There are no further collapses.

haviourally correct learners cope rather well with such memory restrictions [13,7]. This marks the most important and surprising insight of this work. We provide the necessary results in Section 3.2 and collect our findings in the lucid Figure 1(b).

2 Preliminaries

2.1 Language Learning in the Limit

In this section, we discuss the notation used and the system for learning criteria which we follow [15]. Notation which is not introduced follows the textbook [20].

Starting with the mathematical notation, we use $\subsetneq$ and $\subseteq$ to denote the proper subset and subset relation between sets, respectively. We denote with $\mathbb{N} = \{0, 1, 2, \dots\}$ the set of all natural numbers. With $\emptyset$ and ε we denote the *empty set* and *empty string*, respectively. Furthermore, we let $\mathcal{P}$ and $\mathcal{R}$ be the set of all partial and total computable functions $p\colon \mathbb{N} \to \mathbb{N}$, respectively. We fix an effective numbering $\{\varphi_e\}_{e\in\mathbb{N}}$ of $\mathcal{P}$ and denote with $W_e = \mathrm{dom}(\varphi_e)$ the e-th computably enumerable set. This way, we interpret the natural number e as an *index* or *hypothesis* for the set W_e. Regarding important computable functions, we fix with $\langle .,. \rangle$ a computable coding function. We use π_1 and π_2 to recover the first and second component, respectively. Furthermore, we write pad for an injective computable function such that, for all $e, k \in \mathbb{N}$, we have $W_e = W_{\mathrm{pad}(e,k)}$. We use unpad_1 and unpad_2 to compute the first and second component of $\mathrm{pad}(.,.)$, respectively. Note that both functions can be extended iteratively to more coordinates. Lastly, we let ind compute an index for any given finite set.

We aim to learn *languages*, that is, recursively enumerable sets $L \subseteq \mathbb{N}$. These will be learned by *learners* which are partial computable functions. By $\#$ we denote the

pause symbol and for any set S we denote $S_\# \coloneqq S \cup \{\#\}$. Furthermore, a *text* is a total function $T\colon \mathbb{N} \to \mathbb{N} \cup \{\#\}$, the collection of all texts we denote with $\mathbf{Txt}$. For any text or sequence T, we let $\mathrm{content}(T) \coloneqq \mathrm{range}(T) \setminus \{\#\}$ be the *content* of T. A text of a language L is such that $\mathrm{content}(T) = L$, the collection of all texts of L we denote with $\mathbf{Txt}(L)$. For $n \in \mathbb{N}$, we denote by $T[n]$ the initial sequence of T of length n, that is, $T[0] \coloneqq \varepsilon$ and $T[n] \coloneqq (T(0), T(1), \ldots, T(n-1))$. For a set S, we call the text where all elements of S are presented in strictly increasing order (followed by infinitely many pause symbols if S is finite) the *canonical text of* S. Furthermore, we call the sequence of all elements of S presented in strictly ascending order the *canonical sequences of* S. On finite sequences we use $\subseteq$ to denote the *extension relation* and $\leq$ to denote the order on sequences interpreted as natural numbers. Furthermore, for tuples of finite sets and numbers (D, t) and (D', t'), we define the order $\preceq$ such that $(D, t) \preceq (D', t')$ if and only if $t \leq t'$ and there exists a text T such that $D = \mathrm{content}(T[t])$ and $D' = \mathrm{content}(T[t'])$. In addition, given two sequences σ and τ we write $\sigma^\frown\tau$ to denote the concatenation of these. Occasionally, we omit writing $^\frown$ for readability.

We formalise learning criteria using the following system [15]. An *interaction operator* β is given a learner $h \in \mathcal{P}$ and a text $T \in \mathbf{Txt}$ and outputs a (partial) function p. Intuitively, β provides the information for the learner to make its guesses. We consider the interaction operators $\mathbf{G}$ for *Gold-style* or *full-information* learning [9], $\mathbf{Psd}$ for *partially set-driven* or *rearrangement-independent* learning [2,21] and $\mathbf{Sd}$ for *set-driven* learning [22]. Define, for any $i \in \mathbb{N}$,

$$\begin{aligned}\mathbf{G}(h, T)(i) &\coloneqq h(T[i]),\\ \mathbf{Psd}(h, T)(i) &\coloneqq h(\mathrm{content}(T[i]), i),\\ \mathbf{Sd}(h, T)(i) &\coloneqq h(\mathrm{content}(T[i])).\end{aligned}$$

Intuitively, Gold-style learners have full information on the elements presented to them. Partially set-driven learners, however, base their guesses on the total amount of elements presented and the content thereof. Lastly, set-driven learners only base their conjectures on the content given to them. Furthermore, for any β-learner h, we write h^* for its starred learner, that is, the $\mathbf{G}$-learner which simulates h. For example, if $\beta = \mathbf{Sd}$, then, for any sequence σ, $h^*(\sigma) = h(\mathrm{content}(\sigma))$.

When it comes to learning, we can distinguish between various criteria for successful learning. The first such criterion is *explanatory* learning ($\mathbf{Ex}$, [9]). Here, a learner is expected to converge to a single, correct hypothesis in order to learn a language. This can be loosened to require the learner to converge semantically, that is, from some point onwards it must output correct hypotheses which may change syntactically [5,19]. This is referred to as *behaviourally correct* learning ($\mathbf{Bc}$). Formally, a *learning restriction* δ is a predicate on a total learning sequence p, that is, a total function, and a text $T \in \mathbf{Txt}$. For the mentioned criteria we have

$$\begin{aligned}\mathbf{Ex}(p, T) &:\Leftrightarrow \exists n_0 \forall n \geq n_0 \colon p(n) = p(n_0) \wedge W_{p(n_0)} = \mathrm{content}(T),\\ \mathbf{Bc}(p, T) &:\Leftrightarrow \exists n_0 \forall n \geq n_0 \colon W_{p(n)} = \mathrm{content}(T).\end{aligned}$$

These success criteria can be expanded in order to model natural learning restrictions. Our focus lies on monotonic learners [11,23]. *Strongly monotone* learning ($\mathbf{SMon}$)

forms the basis. Here, the learner may never discard elements which were once present in its previous hypotheses. This restrictive criterion can be loosened to hold only on the elements of the target language, that is, the learner may never discard such elements from the language which it already proposed in previous hypotheses. This is referred to as *monotone* learning ($\mathbf{Mon}$). This is formalized as

$$\mathbf{SMon}(p,T) :\Leftrightarrow \forall n,m\colon n \leq m \Rightarrow W_{p(n)} \subseteq W_{p(m)},$$
$$\mathbf{Mon}(p,T) :\Leftrightarrow \forall n,m\colon n \leq m \Rightarrow W_{p(n)} \cap \mathrm{content}(T) \subseteq W_{p(m)} \cap \mathrm{content}(T).$$

Given two restrictions δ and δ', we denote their combination, that is, their intersection, with $\delta\delta'$. Finally, $\mathbf{T}$, the always true predicate, denotes the absence of a restriction.

Now, a *learning criterion* is a tuple $(\alpha, \mathcal{C}, \beta, \delta)$, where $\mathcal{C}$ is a set of admissible learners, typically $\mathcal{P}$ or $\mathcal{R}$, β is an interaction operator and α and δ are learning restrictions. We denote this learning criterion as $\tau(\alpha)\mathcal{C}\mathbf{Txt}\beta\delta$. In the case of $\mathcal{C} = \mathcal{P}$, $\alpha = \mathbf{T}$ or $\delta = \mathbf{T}$ we omit writing the respective symbol. Now, an admissible learner $h \in \mathcal{C}$ $\tau(\alpha)\mathcal{C}\mathbf{Txt}\beta\delta$-learns a language L if and only if on *arbitrary* text $T \in \mathbf{Txt}$ we have $\alpha(\beta(h,T),T)$ and on texts of the target language $T \in \mathbf{Txt}(L)$ we have $\delta(\beta(h,T),T)$. With $\tau(\alpha)\mathcal{C}\mathbf{Txt}\beta\delta(h)$ we denote the class of languages $\tau(\alpha)\mathcal{C}\mathbf{Txt}\beta\delta$-learned by h and with $[\tau(\alpha)\mathcal{C}\mathbf{Txt}\beta\delta]$ we denote the set containing, for all $h' \in \mathcal{C}$, all classes $\tau(\alpha)\mathcal{C}\mathbf{Txt}\beta\delta(h')$. Note that restrictions which hold globally (that is, on arbitrary text) are denoted using $\tau(.)$.

2.2 Normal Forms in Inductive Inference

The introduced learning restrictions all fall into the scope of delayable restrictions. Informally, the hypotheses of a delayable restriction may be postponed arbitrarily but not indefinitely. Formally, we call a learning restriction δ *delayable* if and only if for all texts T and T' with $\mathrm{content}(T) = \mathrm{content}(T')$, all learning sequences p and all total, unbounded non-decreasing functions r, we have that if $\delta(p,T)$ and, for all n, $\mathrm{content}(T[r(n)]) \subseteq \mathrm{content}(T'[n])$, then $\delta(p \circ r, T')$. Furthermore, we call a restriction *semantic* if and only if for any learning sequences p and p' and any text T, we have that if $\delta(p,T)$ and, for all n, $W_{p(n)} = W_{p'(n)}$, then $\delta(p',T)$. Intuitively, a restriction is semantic if any hypothesis could be replaced by a semantically equivalent one without violating the learning restriction. Note that all mentioned restrictions are delayable and all except for $\mathbf{Ex}$ are semantic. In particular, one can provide general results when talking about delayable or semantic restrictions.

Theorem 1 ([12]; [14]). *For any interaction operator β, delayable restriction δ and semantic restriction δ', we have $[\mathcal{R}\mathbf{TxtG}\delta] = [\mathbf{TxtG}\delta]$ and $[\mathcal{R}\mathbf{Txt}\beta\delta'] = [\mathbf{Txt}\beta\delta']$.*

3 Studying Monotone Learning Restrictions

We investigate monotone learners imposed with various restrictions and compare them to their strongly monotone counterpart. We split this study into two parts, first studying explanatory learners in Section 3.1 and then behaviourally correct ones in Section 3.2.

Before we dive into the respective part, we note that it is a well-established fact that strongly monotone learners are significantly weaker than their monotone counterpart. In particular, the class $\mathcal{L} = \{2\mathbb{N}\} \cup \{\{0, 2, 4, \ldots, 2k, 2k+1\} \mid k \in \mathbb{N}\}$ is learnable by a $\mathbf{TxtSdMonEx}$-learner, however, any $\mathbf{TxtGSMonBc}$-learner fails to do so. We remark that the separating class can also be learned by a total monotone learner.

Theorem 2. *We have* $[\mathcal{R}\mathbf{TxtSdMonEx}] \setminus [\mathbf{TxtGSMonBc}] \neq \emptyset$.

Despite this fundamental separation, we observe similarities between monotone and strongly monotone explanatory learners. These similarities are not only reflected by the overall pairwise relation of the different settings, but also by the techniques used to obtain these relations. The main difficulty thereby is to reason why the elements used to contradict strongly monotone learning suddenly are part of a learnable language and, thus, also contradict monotone learning. Furthermore, in order to show strong results, all of these adaptations have to be done while maintaining the original learnability by some strongly monotone learner.

These similarities culminate in Theorem 3, where we show globally monotone learners to be equally powerful as globally strongly monotone ones. This result also holds true when requiring semantic convergence. However, as monotone learners may discard elements from their guesses, the strategy of keeping all once suggested elements regardless of the order (as for strongly monotone learners [13]) is not fruitful for monotone learners. On the contrary, we show that such an equality cannot be obtained. In particular, in Theorem 7 we show that partially set-driven learners are strictly less powerful than their Gold-style counterpart, an unusual result as we discuss in Section 3.2.

3.1 Explanatory Monotone Learning

In this section, we investigate monotone learners when requiring syntactic convergence and also compare them to their strongly monotone counterpart. Building on the thorough discussion of strongly monotone learners [13], we show that the general behaviour of both types of learners is alike. This can be seen, firstly, in the resulting overall picture of the pairwise relations and, secondly, in the way these results are obtained.

Our first result is already a good indication towards how similar these restrictions are. We show that requiring both restrictions to hold globally results in equal learning power. To motivate the idea, note that monotone learners exhibit a strongly monotone behaviour on target languages. If now the learner is required to be monotone on any possible set, as required by global restrictions, it is already globally strongly monotone. Note that this equality, in fact, holds on the level of the restrictions itself.

Theorem 3. *For all restrictions δ and all interaction operators β we have*

$$[\tau(\mathbf{SMon})\mathbf{Txt}\beta\delta] = [\tau(\mathbf{Mon})\mathbf{Txt}\beta\delta].$$

Proof. The inclusion $[\tau(\mathbf{SMon})\mathbf{Txt}\beta\delta] \subseteq [\tau(\mathbf{Mon})\mathbf{Txt}\beta\delta]$ is immediate. For the other inclusion, let h^* be a $\tau(\mathbf{Mon})\mathbf{Txt}\beta\delta$-learner in its starred form. Assume that h^*

is not $\tau(\mathbf{SMon})$. Then, there exists some text T, $i < j$ and x such that $x \in W_{h^*(T[i])} \setminus W_{h^*(T[j])}$. Considering the text $T' \coloneqq T[j]\frown x\frown T(j)\frown T(j+1)\frown \cdots$, we have

$$x \in W_{h^*(T[i])} \cap \operatorname{content}(T') \setminus W_{h^*(T[j])} \cap \operatorname{content}(T').$$

Thus, h^* is not $\tau(\mathbf{Mon})$ on text T', a contradiction. □

In particular, this implies that all separations and equalities known for globally strongly monotone learners also hold for globally monotone ones. Most notably, Gold-style globally monotone learners are strictly less powerful than their total counterpart.

Gold-style monotone learners, being delayable, can be assumed *total* without loss of learning power [12]. We show that these learners are more powerful than their partially set-driven counterpart. In particular, we show that even strongly monotone Gold-style learners are more powerful than any partially set-driven monotone learner. We do so by learning a class of languages on which the learner, in order to discard certain elements, needs to know the order the information appeared in. This, no partially set-driven monotone learner can do.

Theorem 4. *We have* $[\mathbf{TxtGSMonEx}] \setminus [\mathbf{TxtPsdMonEx}] \neq \emptyset$.

Next, we show that a partial learner, even sustaining a severe memory restriction and expected to be strongly monotone, is still more powerful than any total monotone partially set-driven learner. When constructing a separating class of languages, the partial learner simply awaits the guess of the total learner to, then, learn a different language.

Theorem 5. *We have* $[\mathbf{TxtSdSMonEx}] \setminus [\mathcal{R}\mathbf{TxtPsdMonEx}] \neq \emptyset$.

Proof. We adapt the proof of the separation from total $\mathbf{SMon}$-learners [13, Thm. 11] as follows. Let $h \in \mathcal{P}$ be the following learner. With p_0 being such that $W_{p_0} = \emptyset$, let for each finite set $D \subseteq \mathbb{N}$

$$h(D) = \begin{cases} p_0, & \text{if } D = \emptyset, \\ \operatorname{ind}(D), & \text{else, if } |D| = 1, \\ \uparrow, & \text{else, if } \exists x \in D\colon \varphi_x(0)\uparrow \vee \operatorname{unpad}_2(\varphi_x(0)) \notin \{1,2\}, \\ e, & \text{else, if } \forall x \in D\colon \operatorname{unpad}_1(\varphi_x(0)) = e, \\ e', & \text{else, if} \\ & \quad \big(\exists y \forall x \in D\colon \operatorname{unpad}_2(\varphi_x(0)) = 1 \Rightarrow \operatorname{unpad}_1(\varphi_x(0)) = y\big) \wedge \\ & \quad \wedge \big(\forall x \in D\colon \operatorname{unpad}_2(\varphi_x(0)) = 2 \Rightarrow \operatorname{unpad}_1(\varphi_x(0)) = e'\big), \\ \uparrow, & \text{otherwise.} \end{cases}$$

The intuition is the following. While no elements are presented, h conjectures (a code for) the empty set. Once, a single element is presented, h suggests (a code for) that singleton. Thus, h learns all singletons. Given more elements, h either outputs the first coordinate of the elements (if they all coincide), or another code if there are different second coordinates. In case of equal second coordinates but different first coordinates, h is undefined.

Let $\mathcal{L} = \mathbf{TxtSdSMonEx}(h)$. Assume there exists a $\mathcal{R}\mathbf{TxtPsdMonEx}$-learner h' which learns $\mathcal{L}$, that is, $\mathcal{L} \subseteq \mathcal{R}\mathbf{TxtPsdMonEx}(h')$. Since h learns all singletons, so does h'. Thus, there is a total, strictly monotone function $t \in \mathcal{R}$ such that $t(0) > 0$ and for each x

$$x \in W_{h'(\{x\},t(x))}. \tag{1}$$

With **ORT** ([3]), we get a total recursive predicate $P \in \mathcal{R}$, a strictly monotone increasing $a \in \mathcal{R}$ and indices $e, e' \in \mathbb{N}$ such that for all $i \in \mathbb{N}$, using $\tilde{t}(i) \coloneqq \sum_{j=0}^{i} t(a(j)) + j$ as abbreviation,

$$\begin{aligned} P(i) &\Leftrightarrow h'(\operatorname{content}(a[i]), \tilde{t}(i)) \neq h'(\operatorname{content}(a[i+1]), \tilde{t}(i)+1), \\ W_e &= \{a(i) \mid \forall j \leq i \colon P(j)\}, \\ W_{e'} &= \{a(i) \mid \forall j < i \colon P(j)\}, \\ \varphi_{a(i)}(0) &= \begin{cases} \operatorname{pad}(e,1), & \text{if } P(i), \\ \operatorname{pad}(e',2), & \text{otherwise.} \end{cases} \end{aligned}$$

We show that W_e and $W_{e'}$ are in $\mathcal{L}$.

1. Case: W_e is infinite. This means for all i we have $P(i)$. Thus, $W_e = W_{e'}$. Thus, it suffices to show $W_e \in \mathcal{L}$. Let $T \in \mathbf{Txt}(W_e)$. For $n > 0$, let $D_n \coloneqq \operatorname{content}(T[n])$. As long as $D_n = \emptyset$, we have $h(D_n) = p_0$, i.e. a code for the empty set. When $|D_n| = 1$, we have $h(D_n) = \operatorname{ind}(D_n)$, a code for the singleton D_n. Once D_n contains more than one element, $h(D_n)$ starts unpadding. As, for all i, $\varphi_{a(i)}(0) = \operatorname{pad}(e,1)$, we have $\operatorname{unpad}_1(\{\varphi_x(0) \mid x \in D_n\}) = \{e\}$. Thus, h is strongly monotone and will output e correctly.
2. Case: W_e is finite. Let k be such that $W_e = \{a(j) \mid j < k\}$ and $W_{e'} = \{a(j) \mid j < k+1\}$. Again, as long as no elements or only one element is shown, h will output a code for the empty, respectively singleton set. As $W_e \subseteq W_{e'}$ and $\operatorname{unpad}_1(\{\varphi_x(0) \mid x \in W_e\}) = \{e\}$, h will output e as long as it sees only elements from W_e. Once it sees $a(k) \in W_{e'}$, it correctly changes its mind to e'. This maintains strong monotonicity and is the correct behaviour.

Thus, $W_e, W_{e'} \in \mathcal{L}$. We show that h' cannot learn both simultaneously.

1. Case: W_e is infinite. On the text $a(0)^{t(a(0))}a(1)^{t(a(1))+1}a(2)^{t(a(2))+2}\ldots$ of W_e, the learner h' makes infinitely many mind changes. Thus, it cannot learn W_e, a contradiction.
2. Case: W_e is finite. Let k be minimal such that $\neg P(k)$, and thus $W_e = \operatorname{content}(a[k])$ and $W_{e'} = \operatorname{content}(a[k+1])$. By Condition (1) and monotonicity of h' on $W_{e'}$ we have $a(k) \in W_{h'(\operatorname{content}(a[k+1]),\tilde{t}(k)+1)}$, as $a(k)^{\tilde{t}(k)\frown}a[k]$ is a sequence of elements in $W_{e'}$ and $a(k) \in W_{e'}$. Since $\neg P(k)$, we get $h'(\operatorname{content}(a[k]), \tilde{t}(k)) = h'(\operatorname{content}(a[k+1]), \tilde{t}(k)+1)$ and, thus, $a(k) \in W_{h'(\operatorname{content}(a[k]),t(a(k))+k)}$. For each $t \geq \tilde{t}(k)$, we have that $(\operatorname{content}(a[k]), t)$ is an initial sequence for some text of $W_{e'}$, and thus, by monotonicity of h' we get $a(k) \in W_{h'(\operatorname{content}(a[k]),t)}$. As $a(k) \notin W_e = \operatorname{content}(a[k])$, h' cannot identify W_e, a contradiction. □

To complete Figure 1(a), it remains to be shown that globally strongly monotone partially set-driven learners are more powerful than their monotone set-driven counterpart. The separation from strongly monotone set-driven learners has already been shown [13]. We provide a self-learning class [4] to show that globally strongly monotone partially set-driven learners outperform *unrestricted* set-driven learners. This result emphasises the weakness of set-driven learners which results from a lack of "learning time" [8].

We note that, when studying learners which may be undefined even on input belonging to a target language, a similar class is used to separate strongly monotone Gold-style learners from total set-driven learners [10].

Theorem 6. *We have* $[\tau(\mathbf{SMon})\mathbf{TxtPsdEx}] \setminus [\mathbf{TxtSdEx}] \neq \emptyset$.

3.2 Behaviourally Correct Monotone Learning

In this section we consider an analogous question: How do monotone and strongly monotone learners interact when requiring semantic convergence? By Theorem 3 and the findings of Kötzing and Schirneck [13], we already have that globally monotone set-driven (and even Gold-style) learners are as powerful as strongly monotone Gold-style learners. The mentioned learners are, due to Theorem 2, less powerful than total set-driven monotone ones. This, in particular, implies that a "complete collapse" of the learning considered criteria as for strongly monotone learners [13] is impossible. As partially set-driven monotone (explanatory) learners are more powerful than set-driven behaviourally correct ones [14], only one question remains, namely, whether Gold-style $\mathbf{Mon}$-learners may be separated from partially set-driven $\mathbf{Mon}$-learners? Studies of various other restrictions [13,7], show that behaviourally correct partially set-driven learners are often as powerful as their respective Gold-style counterpart.

Surprisingly, for monotone behaviourally correct learners, such an equality does *not* hold, as we show with the next result. The idea is to construct a class of languages where the learner must keep track of the order the elements were presented in, in order to safely discard them at a later point in learning-time. To obtain this result, we apply the technique of self-learning classes [4] using the Operator Recursion Theorem [3]. Note that this result already completes Figure 1(b), as monotone $\mathbf{Bc}$-learners may be assumed total [14].

Theorem 7. *We have* $[\mathbf{TxtGMonEx}] \setminus [\mathbf{TxtPsdMonBc}] \neq \emptyset$.

Proof. We provide a class witnessing the separation using self-learning classes [4, Thm. 3.6]. Consider the learner which for a finite sequence σ is defined as

$$h(\sigma) = \begin{cases} \mathrm{ind}(\emptyset), & \text{if } \mathrm{content}(\sigma) = \emptyset, \\ \varphi_{\max(\mathrm{content}(\sigma))}(\sigma), & \text{otherwise.} \end{cases}$$

Let $\mathcal{L} = \mathbf{TxtGMonEx}(h)$. Assume there exists a $\mathbf{TxtPsdMonBc}$-learner h' which learns $\mathcal{L}$, that is, $\mathcal{L} \subseteq \mathbf{TxtPsdMonBc}(h')$. By the Operator Recursion Theorem (**ORT**, [3]), there exists a family of strictly monotone increasing, total computable functions $(a_j)_{j\in\mathbb{N}}$ with pairwise disjoint range, a total computable function $f \in \mathcal{R}$, an index

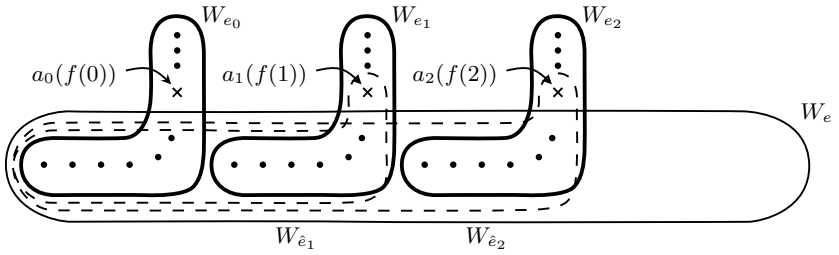


Fig. 2: A depiction of the class $\mathcal{L}'$. Given j, the dashed line depicts the set $W_{\hat{e}_j}$ and the cross indicates the element $a_j(f(j))$.

$e \in \mathbb{N}$ and two families of indices $(e_j)_{j\in\mathbb{N}}, (\hat{e}_k)_{k\in\mathbb{N}}$ such that for all finite sequences σ, where $\mathrm{first}(\sigma)$ is the first non-pause element in the sequence σ, we have

$$\varphi_{a_j(i)}(\sigma) = \begin{cases} e_j, & \text{if } \mathrm{content}(\sigma) \subseteq \mathrm{range}(a_j), \\ \hat{e}_k, & \text{else, if } \exists k\colon a_k(f(k)) \in \mathrm{content}(\sigma) \vee \\ & \exists k\colon \mathrm{first}(\sigma) \in \mathrm{range}(a_k) \wedge \\ & \quad \wedge \max\{j \mid \mathrm{content}(\sigma) \cap \mathrm{range}(a_j) \neq \emptyset\} = k, \\ e, & \text{otherwise.} \end{cases}$$

$$f(j) = \text{first } i \text{ found such that } a_j(i) \in W_{h'(\mathrm{content}(a_j[i]),i)},$$

$$W_{e_j} = \mathrm{range}(a_j),$$

$$W_{\hat{e}_k} = \bigcup_{j'\leq k} \mathrm{content}(a_{j'}[f(j')]) \cup \{a_k(f(k))\},$$

$$W_e = \bigcup_j \mathrm{content}(a_j[f(j)]).$$

Let $\mathcal{L}' = \{W_{e_j} \mid j \in \mathbb{N}\} \cup \{W_{\hat{e}_k} \mid k > 0\} \cup \{W_e\}$. Figure 2 shows a depiction of the class $\mathcal{L}'$. We show that $\mathcal{L}'$ can be learned by h, but not by h'. The intuition is the following. For some j, as long as only elements from W_{e_j} are presented, h will suggest e_j as its hypothesis. Thus, h' needs to learn W_{e_j} as well and eventually overgeneralize, that is, at some point i we have $\mathrm{content}(a_j[i]) \subsetneq W_{h'(\mathrm{content}(a_j[i]),i)}$. The function $f(j)$ finds such i. Once the overgeneralization happens, the text presents, for $j' \neq j$, elements from $\mathrm{range}(a_{j'})$. Knowing the order in which the elements were presented, the learner h now either keeps or discards the element $a_j(f(j))$ in its next hypothesis depending whether $j' < j$ or $j < j'$, respectively. If $j' < j$, h needs to keep $a_j(f(j))$ in its hypothesis as it still may be presented the set $W_{\hat{e}_j}$. Otherwise, it suggests the set W_e, only changing its mind if it sees, for appropriate $i \in \mathbb{N}$, an element of the form $a_i(f(i))$. Then, h is certain to be presented $W_{\hat{e}_i}$. So the full-information learner h can deal with this new information and preserve monotonicity, while h' cannot, as it does not know which information came first.

We proceed with the formal proof that h $\mathbf{TxtGMonEx}$-learns $\mathcal{L}'$. Let $L' \in \mathcal{L}'$ and $T' \in \mathbf{Txt}(L')$. We first show the $\mathbf{Ex}$-convergence and the monotonicity afterwards. For the former, we distinguish the following cases.

1. Case: For some j, we have $L' = W_{e_j}$. Let n_0 be such that $\mathrm{content}(T'[n]) \neq \emptyset$. Then, for $n \geq n_0$, there exists some i such that $a_j(i) = \max(\mathrm{content}(T'[n]))$. Thus,

$$h(T'[n]) = \varphi_{\max(\mathrm{content}(T'[n]))}(T'[n]) = \varphi_{a_j(i)}(T'[n]) = e_j.$$

   Hence, h learns W_{e_j} correctly.
2. Case: We have $L' = W_e$. Let $n_0 \in \mathbb{N}$ be the minimal and let $k_0 \in \mathbb{N}$ be such that $\mathrm{content}(T'[n_0]) \neq \emptyset$ and $\mathrm{first}(T'[n_0]) \in \mathrm{range}(a_{k_0})$. Let $n_1 \geq n_0$ be minimal such that there exists $k > k_0$ such that $\mathrm{content}(T'[n_1])$ also contains elements from $\mathrm{content}(a_k)$. Then, for $n > n_1$ we have that $h(T'[n]) = e$, as there exists no j with $a_j(f(j)) \in \mathrm{content}(T')$ and also $\max\{j \mid \mathrm{content}(T'[n]) \cap \mathrm{range}(a_j) \neq \emptyset\} \neq k_0$. Thus, h learns W_e correctly.
3. Case: For some $k > 0$ we have $L' = W_{\hat{e}_k}$. In this case, there exists n_0 such that, for some $k' < k$, $\mathrm{range}(a_{k'}) \cap \mathrm{content}(T'[n_0]) \neq \emptyset$ and $a_k(f(k)) \in \mathrm{content}(T'[n_0])$. Then, for $n \geq n_0$, we have $h(T'[n]) = \hat{e}_k$. Therefore, h learns $W_{\hat{e}_k}$ correctly.

We show that the learning is monotone. Let $n \in \mathbb{N}$. As long as $\mathrm{content}(T'[n])$ is empty, h returns $\mathrm{ind}(\emptyset)$. Once $\mathrm{content}(T'[n])$ is not empty anymore and as long as $\mathrm{content}(T'[n])$ only contains elements from, for some j, $\mathrm{range}(a_j)$, the learner h outputs (a code for) the set W_{e_j}. Note that j is the index of the element $\mathrm{first}(T'[n])$, that is, $\mathrm{first}(T'[n]) \in \mathrm{range}(a_j)$. If ever, for some later n, $\mathrm{content}(T'[n]) \setminus \mathrm{range}(a_j) \neq \emptyset$, then h only changes its mind if there exists $k > j$ such that $\mathrm{content}(T'[n]) \cap \mathrm{range}(a_k) \neq \emptyset$ (note that in case $j < k$, h does not change its mind). Depending on whether $a_k(f(k)) \in \mathrm{content}(T'[n])$ or not, h changes its mind to (a code of) either $W_{\hat{e}_k}$ or W_e, respectively. In the former case, the learner h is surely presented the set $W_{\hat{e}_k}$, making this mind change monotone. In the latter case, no element of $W_{e_j} \setminus \mathrm{content}(a_j[f(j)])$ is contained the target language. These are exactly the elements h discards from its hypothesis, keeping a monotone behaviour. The learner only changes its mind again if it witnesses, for some $k' \geq k$, the element $a_{k'}(f(k'))$. It will then output (a code of) the set $W_{\hat{e}_{k'}}$. This is, again, monotonic behaviour, as h is sure to be presented the set $W_{\hat{e}_{k'}}$. Altogether, h is monotone on any text of L'.

Thus, h identifies all languages in $\mathcal{L}'$ correctly. Now, we show that h' cannot do so too. We do so by providing a text of W_e where h' makes infinitely many wrong guesses. To that end, consider the text T of W_e given as $a_0[f(0)]a_1[f(1)]a_2[f(2)]\ldots$ For $j > 0$, since $a_j(f(j)) \in W_{h'(\mathrm{content}(a_j[f(j)]),f(j))}$, we have

$$a_j(f(j)) \in W_{h'(\mathrm{content}(T[\sum_{m \leq j} f(m)]),\sum_{m \leq j} f(m))},$$

as $T[\sum_{m \leq j} f(m)]$ is an initial sequence for a text for $W_{\hat{e}_j}$. But, since $a_j(f(j)) \notin W_e$, h' makes infinitely many incorrect conjectures and thus does not identify W_e on the text T correctly, a contradiction. □

Acknowledgements

We would like to thank the anonymous reviewers for their helpful suggestions and comments. We believe that their feedback helped improve this work.

References

1. Angluin, D.: Inductive inference of formal languages from positive data. Information and Control **45**, 117–135 (1980)
2. Blum, L., Blum, M.: Toward a mathematical theory of inductive inference. Information and Control **28**, 125–155 (1975)
3. Case, J.: Periodicity in generations of automata. Mathematical Systems Theory **8**, 15–32 (1974)
4. Case, J., Kötzing, T.: Strongly non-U-shaped language learning results by general techniques. Information and Computation **251**, 1–15 (2016)
5. Case, J., Lynes, C.: Machine inductive inference and language identification. In: Proc. of the International Colloquium on Automata, Languages and Programming (ICALP). pp. 107–115 (1982)
6. Doskoč, V., Kötzing, T.: Mapping monotonic restrictions in inductive inference. CoRR (2020)
7. Doskoč, V., Kötzing, T.: Cautious limit learning. In: Proc. of the International Conference on Algorithmic Learning Theory (ALT) (2020)
8. Fulk, M.A.: Prudence and other conditions on formal language learning. Information and Computation **85**, 1–11 (1990)
9. Gold, E.M.: Language identification in the limit. Information and Control **10**, 447–474 (1967)
10. Jain, S.: Strong monotonic and set-driven inductive inference. J. Exp. Theor. Artif. Intell. **9**, 137–143 (1997)
11. Jantke, K.: Monotonic and non-monotonic inductive inference. New Generation Computing **8**, 349–360 (1991)
12. Kötzing, T., Palenta, R.: A map of update constraints in inductive inference. Theoretical Computer Science **650**, 4–24 (2016)
13. Kötzing, T., Schirneck, M.: Towards an atlas of computational learning theory. In: Proc. of the Symposium on Theoretical Aspects of Computer Science (STACS). pp. 47:1–47:13 (2016)
14. Kötzing, T., Schirneck, M., Seidel, K.: Normal forms in semantic language identification. In: Proc. of the International Conference on Algorithmic Learning Theory (ALT). pp. 76:493–76:516 (2017)
15. Kötzing, T.: Abstraction and Complexity in Computational Learning in the Limit. Ph.D. thesis, University of Delaware (2009)
16. Lange, S., Zeugmann, T.: Monotonic versus non-monotonic language learning. In: Nonmonotonic and Inductive Logic. pp. 254–269 (1993)
17. Lange, S., Zeugmann, T.: Set-driven and rearrangement-independent learning of recursive languages. Mathematical Systems Theory **29**, 599–634 (1996)
18. Lange, S., Zeugmann, T., Kapur, S.: Monotonic and dual monotonic language learning. Theoretical Computer Science **155**, 365–410 (1996)
19. Osherson, D.N., Weinstein, S.: Criteria of language learning. Information and Control **52**, 123–138 (1982)
20. Rogers Jr., H.: Theory of recursive functions and effective computability. Reprinted by MIT Press, Cambridge (MA) (1987)
21. Schäfer-Richter, G.: Über Eingabeabhängigkeit und Komplexität von Inferenzstrategien. Ph.D. thesis, RWTH Aachen University, Germany (1984)
22. Wexler, K., Culicover, P.W.: Formal principles of language acquisition. MIT Press, Cambridge (MA) (1980)
23. Wiehagen, R.: A thesis in inductive inference. In: Nonmonotonic and Inductive Logic. pp. 184–207 (1991)