Set-Driven and Rearrangement-Independent Learning of Recursive Languages[1]

Steffen Lange
HTWK Leipzig
FB Mathematik und Informatik
PF 30066
04251 Leipzig, Germany
steffen@informatik.th-leipzig.de

Thomas Zeugmann
Department of Informatics
Graduate School of Inf. Sci. and EE
Kyushu University
Fukuoka 812-81, Japan
thomas@i.kyushu-u.ac.jp

Key words: Algorithmic Learning Theory, Computational issues in A.I., Automata and Formal Languages

[1]This is a substantially revised version of the author's paper at the 5th International Workshop on Algorithmic Learning Theory (ALT'94), Lecture Notes in Artificial Intelligence 872, 453 – 468.

Abstract

The present paper studies the impact of order independence to the learnability of indexed families $\mathcal{L}$ of uniformly recursive languages from positive data. In particular, we consider *set-driven* and *rearrangement-independent* learners, i.e., learning devices whose output exclusively depends on the range and on the range and length of their input, respectively. The impact of set-drivenness and rearrangement-independence on the behavior of learners to their learning power is studied in dependence on the *hypothesis space* the learners may use. We distinguish between *exact* learnability ($\mathcal{L}$ has to be inferred with respect to $\mathcal{L}$), *class-preserving* learning ($\mathcal{L}$ has to be inferred with respect to some suitably chosen enumeration of all the languages from $\mathcal{L}$), and *class-comprising* inference ($\mathcal{L}$ has to be learned with respect to some suitably chosen enumeration of uniformly recursive languages containing at least all the languages from $\mathcal{L}$).

Furthermore, we consider the influence of set-drivenness and rearrangement-independence for learning devices that realize the *subset principle* to different extents. Thereby we distinguish between *strong-monotonic, monotonic* and *weak-monotonic* or *conservative* learning.

The results obtained are threefold. First, rearrangement-independent learning does not constitute a restriction except in the case of monotonic learning. Next, we prove that for all but two of the considered learning models set-drivenness is a severe restriction. However, class-comprising set-driven *conservative* learning is exactly as powerful as unrestricted class-comprising *conservative* learning. Finally, the power of class-comprising set-driven learning in the limit is characterized by equating the collection of learnable indexed families with the collection of class-comprisingly conservatively inferable indexed families. These results considerably extend previous work done in the field (cf., e.g., Schäfer-Richter [20] and Fulk [5]).

1. Introduction

Language acquisition is one of the main topics in cognitive science, epistemology, linguistic and psycholinguistic theory as well as of machine learning and algorithmic learning theory. All these disciplines share the common goal to gain a better understanding of what learning really is. This goal is of special interest to computer science if a learning computer should not remain a fiction.

Formal language learning may be characterized as the study of systems that map evidence on a language into hypotheses about it. Of special interest is the investigation of scenarios in which the sequence of hypotheses *stabilizes* to an *accurate* and *finite* description (a grammar) of the target language. Clearly, then some form of learning must have taken place. In his pioneering paper, Gold [6] gave precise definitions of the concepts "evidence," "stabilization," and "accuracy" resulting in the model of learning in the limit. During the last decades, Gold-style formal language learning has attracted a lot of attention by computer scientists (cf., e.g., Osherson, Stob and Weinstein [19] and the references therein). Most of the work done in the field has been aimed at the following goals: showing what general collections of language classes are learnable, characterizing those collections of language classes that can be learned, studying the impact of several postulates on the behavior of learners to their learning power, and dealing with the influence of various parameters to the efficiency of learning.

In this paper we aim to investigate the learning capabilities of learners that fulfill *simultaneously various combinations* of desirable properties. For the purpose of motivation and discussion of our research, we introduce some notations. A *text* of a language L is an infinite sequence of strings that eventually contains all strings of L. An algorithmic learner, henceforth called *inductive inference machine* (abbr. IIM), takes as input initial segments of a text, and outputs, from time to time, a hypothesis about the target language. The set $\mathcal{G}$ of all admissible hypotheses is called *hypothesis space*. Furthermore, the sequence of hypotheses has to converge to a hypothesis correctly describing the language to be learned, i.e., after some point, the IIM stabilizes to an accurate hypothesis. If there is an IIM that learns a language L from all texts for it, then L is said to be *learnable in the limit* with respect to the hypothesis space $\mathcal{G}$ (cf. Definition 1).

A first question directly arising when dealing with learning in the limit is whether or not the *order* of information presentation does really influence the capabilities of IIMs. An IIM is said to be *set-driven*, if its output only depends on the *range* of its input. Surprisingly enough, Schäfer-Richter [20] and Fulk [5] proved that set-driven IIMs are less powerful than unrestricted ones. Intuitively, the weakness of set-driven IIMs is caused by the difficulties of handling both, finite and infinite languages. A natural weakening of set-drivenness is rearrangement-independence. An IIM is called *rearrangement-independent* if its output only depends on the *range* and *length* of its input. As it turned out, any collection of languages that can be learned in the limit may also be learned by a rearrangement-independent IIM (cf. Schäfer-Richter [20], Fulk [5]). However, the weakness of set-driven IIMs has been proved in the setting of learning recursively enumerable languages. And indeed, the examples of Schäfer-Richter [20] and Fulk [5] witnessing the weakness of set-driven IIMs are not enumerable families of nonempty uniformly recursive languages, henceforth called *indexed families*. This might lead to the impression that this result is of restricted practical relevance, since numerous potential applications of learning involve indexed families.

Therefore, we study the power of set-driven IIMs in a more realistic setting with respect to potential applications, i.e., we deal exclusively with indexed families. Well-known examples of indexed families are the set of all context sensitive languages, the set of all context free languages as well as the set of all regular languages (cf., e.g., Hopcroft and Ullman [7]). Another famous example for an indexed family is the collection of all pattern languages (cf. Angluin [1]). Although this indexed family contains finite and infinite languages, Lange and Wiehagen [9] succeeded in designing a set-driven IIM learning it (cf. [26, Theorem 2]). Consequently, it is only natural to ask whether or not any learnable indexed family may be learned by a set-driven IIM, too.

A major problem that has to be dealt with when learning from text, is to avoid or to detect *overgeneralization* (also called the *subset problem*), i.e., hypotheses that describe proper *supersets* of the target language. The impact of this problem results simply from the fact that a text cannot supply counterexamples to such hypotheses. IIMs that strictly avoid overgeneralized hypotheses are called *conservative* (cf. Definition 6). As it turns out, neither Schäfer-Richter's [20] nor Fulk's [5] transformation of an arbitrary IIM into a rearrangement-independent one preserves conservativeness. Therefore, we study the problem whether or not rearrangement-independence is a severe restriction for conservative learners. However, this problem has its special peculiarities. Namely, when dealing with conservative learning, the choice of the hypothesis space does seriously influence the learnability of indexed families (cf. Lange and Zeugmann [12]). Hence, we have to distinguish between exact learning, class-preserving inference, and class-comprising identification. If an indexed family $\mathcal{L}$ can be learned with respect to the hypothesis space $\mathcal{L}$, then $\mathcal{L}$ is said to be *exactly* learnable. Furthermore, $\mathcal{L}$ is learnable by a *class-preserving* learning algorithm M, if there is a hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ such that any G_j describes a language from $\mathcal{L}$ and M learns $\mathcal{L}$ with respect to $\mathcal{G}$. That means, if one learns class-preservingly, then one has the freedom to change the enumeration as well as the description of the languages from $\mathcal{L}$. Finally, if any hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ comprising the range of $\mathcal{L}$ may be taken by the learning algorithm, then we call it *class-comprising*. In this setting one has the freedom to change the enumeration, the description and to add elements G_k not describing any language from $\mathcal{L}$ to the hypothesis space. However, since membership in $\mathcal{L}$ is uniformly decidable, we restrict ourselves to consider exclusively hypothesis spaces having a uniformly decidable membership problem.

Several authors proposed the so-called *subset principle* to solve the problem of avoiding overgeneralization (cf., e.g., Berwick [4], Wexler [22]). Informally, the subset principle requires the learner to hypothesize a "least" language from the hypothesis space with respect to set inclusion that fits with the data the IIM has read so far. Therefore, we present some formalizations of learning realizing the subset principle to different extents. First, we require the learning algorithm to produce a sequence of hypotheses describing an augmenting chain of languages, i.e., $L(G_j) \subseteq L(G_k)$, if k is hypothesized on an extension of the text segment that led to j (cf. Definition 5, (A)). We call learners behaving thus *strong-monotonic*. Weakening the latter demand leads to *weak-monotonic* learners that are required to behave strong-monotonically as long as they do not receive data contradicting its actual hypothesis. If they receive strings that *provably misclassify* their actual hypothesis, then weak-monotonic learners are allowed to output any hypothesis (cf. Definition 5, (C)). Third, we refine strong-monotonic learning in that we only require $L(G_j) \cap L \subseteq L(G_k) \cap L$. Now, "least" language is interpreted with respect to the intersection with L. This learning model is called *monotonic* inference (cf. Definition 5, (B)). Strong-monotonic and weak-monotonic learning have been introduced by Jantke [8] and monotonic learning goes back to Wiehagen [24]. Subsequently, we have studied their learning capabilities in the setting of learning indexed families (cf. Lange and Zeugmann [11]). Again, the power of all the monotonic learning models heavily depends on the choice of the hypothesis space (cf. Lange and Zeugmann [12]).

In what follows we study the impact of set-drivenness and rearrangement-independence on all the learning models described above in dependence on the hypothesis space. The results obtained prove that rearrangement-independent learning does not constitute a restriction except in case one learns monotonically. These results have been achieved by non-trivial applications of the characterizations of all types of monotonic learning in terms of finite tell-tales (cf. Lange and Zeugmann [10]). Furthermore, we show that set-drivenness cannot be achieved in general. However, class-comprising conservative learning is exactly as powerful as class-comprising set-driven conservative inference. Moreover, we characterize class-comprising set-driven learning in the limit by equating the collection of learnable indexed families with the collection of class-comprisingly conservatively inferable indexed families. We regard these results as a particular answer to the question of how a "natural" learning algorithm may be designed.

2. Preliminaries

By $\mathbb{N} = \{0, 1, 2, ...\}$ we denote the set of all natural numbers. We set $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$. Let $\varphi_0,\ \varphi_1,\ \varphi_2, ...$ denote any fixed ***acceptable programming system*** of all (and only all) partial recursive functions over $\mathbb{N}$, and let $\Phi_0,\ \Phi_1,\ \Phi_2, ...$ be any associated ***complexity measure*** (cf. Machtey and Young [17]). Then φ_k is the partial recursive function computed by program k in the programming system. Furthermore, let $k, x \in \mathbb{N}$. If $\varphi_k(x)$ is defined (abbr. $\varphi_k(x)\downarrow$) then we also say that $\varphi_k(x)$ converges; otherwise, $\varphi_k(x)$ diverges (abbr. $\varphi_k(x)\uparrow$). By $\langle .,. \rangle \colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ we denote ***Cantor's pairing function***, i.e.,

$\langle x, y\rangle = ((x+y)^2 + 3x + y)/2$ for all $x, y \in \mathbb{N}$.

In what follows we assume familiarity with formal language theory (cf. Hopcroft and Ullman [7]). By Σ we denote any fixed finite alphabet of symbols. Let Σ^* be the free monoid over Σ, and let $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$, where ε denotes the empty string. Any subset $L \subseteq \Sigma^*$ is called a language. Let L be a language and $t = s_0, s_1, s_2, ...$ an infinite sequence of strings from Σ^* such that $range(t) = \{s_k \mid k \in \mathbb{N}\} = L$. Then t is said to be a ***text*** for L or, synonymously, a ***positive presentation***. Let L be a language. By $text(L)$ we denote the set of all positive presentations of L. Moreover, let t be a text and let x be a number. Then, t_x denotes the initial segment of t of length $x+1$, and $t_x^+ =_{df} \{s_k \mid k \leq x\}$.

Next, we introduce the notion of the ***canonical text*** that turned out to be very helpful in proving several theorems. Let L be any nonempty recursive language, and let $s_0, s_1, s_2, ...$ be the lexicographically ordered text of Σ^*. The canonical text of L is obtained as follows. Test sequentially whether $s_z \in L$ for $z = 0, 1, 2, ...$ until the first z is found such that $s_z \in L$. Since $L \neq \emptyset$ there must be at least one z fulfilling the test. Set $t_0 = s_z$. We proceed inductively. For all $x \in \mathbb{N}$ we define:

$$t_{x+1} = \begin{cases} t_x \cdot s_{z+x+1}, & \text{if} \quad s_{z+x+1} \in L, \\ t_x \cdot s, & \text{otherwise, where } s \text{ is the last string in } t_x. \end{cases}$$

In what follows we deal with the learnability of indexed families of uniformly recursive languages defined as follows (cf. Angluin [2]). A sequence $L_0, L_1, L_2, ...$ is said to be an ***indexed family*** $\mathcal{L}$ of uniformly recursive languages provided all L_j are nonempty and there is a recursive function f such that for all numbers j and all strings $s \in \Sigma^*$ we have

$$f(j, s) = \begin{cases} 1, & \text{if} \quad s \in L_j, \\ 0, & \text{otherwise.} \end{cases}$$

In the following we refer to indexed families of uniformly recursive languages as indexed families for short. Note that an indexed family is directly connected with the grammars behind the enumerated languages. In particular, we can consider the indices as compiled grammars (cf. Hopcroft and Ullman [7]). Next, we extend the notion of $text(L)$ to indexed families. Let $\mathcal{L}$ be an indexed family, then we set $text(\mathcal{L}) = \bigcup_{L \in range(\mathcal{L})} text(L)$.

As in Gold [6] we define an ***inductive inference machine*** (abbr. IIM) to be an algorithmic device which works as follows: The IIM takes as its input larger and larger initial segments of a text t and it either requests the next input string, or it first outputs a hypothesis, i.e., a number encoding a certain computer program, and then it requests the next input string.

At this point we specify the semantics of the hypotheses an IIM outputs. For that purpose we have to clarify what hypothesis spaces we choose. We require the inductive inference machines to output indices of grammars, since this learning goal fits well with the intuitive idea of language learning. Furthermore, since we exclusively deal with indexed families $\mathcal{L} = (L_j)_{j\in\mathbb{N}}$ we always take as a space of hypotheses an enumerable family of grammars $G_0, G_1, G_2, ...$ over the terminal alphabet Σ satisfying $range(\mathcal{L}) \subseteq \{L(G_j) \mid j \in \mathbb{N}\}$. Moreover, we require that membership in $L(G_j)$ is uniformly decidable for all $j \in \mathbb{N}$ and all strings $s \in \Sigma^*$. When an IIM outputs a number j, we interpret it to mean that the machine is hypothesizing the grammar G_j. Furthermore, let $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ be any hypothesis space. For notational convenience we use $\mathcal{L}(\mathcal{G})$ to denote $(L(G_j))_{j\in\mathbb{N}}$. Note that $\mathcal{L}(\mathcal{G})$ constitutes itself an indexed family for all hypothesis spaces $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$.

Let t be a text, and $x \in \mathbb{N}$. Then we use $M(t_x)$ to denote the last hypothesis produced by M when successively fed t_x. The sequence $(M(t_x))_{x\in\mathbb{N}}$ is said to ***converge in the limit*** to the number j if and only if either $(M(t_x))_{x\in\mathbb{N}}$ is infinite and all but finitely many terms of it are equal to j, or $(M(t_x))_{x\in\mathbb{N}}$ is nonempty and finite, and its last term is j. Now we define some concepts of learning. We start with learning in the limit.

Definition 1. **(Gold [6])** *Let $\mathcal{L}$ be an indexed family, let L be a language, and let $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ be a hypothesis space.* ***An IIM M CLIM–identifies L from text with respect to $\mathcal{G}$*** *iff for every text t for L, there exists a $j \in \mathbb{N}$ such that the sequence $(M(t_x))_{x\in\mathbb{N}}$ converges in the limit to j and $L = L(G_j)$.*

Furthermore, M CLIM–identifies $\mathcal{L}$ with respect to $\mathcal{G}$ iff, for each $L \in range(\mathcal{L})$, M CLIM–identifies L from text with respect to $\mathcal{G}$.

Finally, let CLIM denote the collection of all indexed families $\mathcal{L}$ for which there are an IIM M and a hypothesis space $\mathcal{G}$ such that M CLIM–identifies $\mathcal{L}$ with respect to $\mathcal{G}$.

Suppose, an IIM identifies some language L. That means, after having seen only finitely many data of L the IIM reached its (unknown) point of convergence and it computed a *correct* and *finite* description of a generator for the target language. Hence, some form of learning must have taken place. Therefore, we use the terms *infer* and *learn* as synonyms for identify.

In the above Definition LIM stands for "limit." Furthermore, the prefix C is used to indicate ***class-comprising*** learning, i.e., the fact that $\mathcal{L}$ may be learned with respect to some hypothesis space comprising $range(\mathcal{L})$. The restriction of $CLIM$ to ***class preserving*** inference is denoted by LIM. That means LIM is the collection of all indexed families $\mathcal{L}$ that can be learned in the limit with respect to a hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ such that $range(\mathcal{L}) = \{L(G_j) \mid j \in \mathbb{N}\}$. Moreover, if a target indexed family $\mathcal{L}$ has to be inferred with respect to the hypothesis space $\mathcal{L}$ itself, then we replace the prefix C by E, i.e., $ELIM$ is the collection of indexed families that can be ***exactly*** learned in the limit. Finally, we adopt this convention in defining all the learning types below.

Moreover, an IIM is required to learn the target language from every text for it. This might lead to the impression that an IIM mainly extracts the range of the information fed to it, thereby neglecting the length and order of the data sequence it reads. IIMs really behaving thus are called set-driven. More precisely, we define:

Definition 2. (Wexler and Culicover, Sec. 2.2, [23]) *Let $\mathcal{L}$ be an indexed family.* ***An IIM M is said to be set-driven*** *with respect to $\mathcal{L}$ iff its output depends only on the range of its input; that is, iff $M(t_x) = M(\hat{t}_y)$ for all $x, y \in \mathbb{N}$, all texts t, $\hat{t} \in text(\mathcal{L})$ provided $t_x^+ = \hat{t}_y^+$.*

Whenever the relevant indexed family $\mathcal{L}$ is clear from the context we refer to set-driven with respect to $\mathcal{L}$ as set-driven for short. Schäfer-Richter [20] as well as Fulk [5], later, and independently proved that set-driven IIMs are less powerful than unrestricted ones. Fulk [5] interpreted the weakening in the learning power of set-driven IIMs by the need of IIMs for time to "reflect" on the input. However, this time cannot be bounded by any *a priori* fixed computable function depending exclusively on the size of the range of the input, since otherwise set-drivenness would not restrict the learning power. Indeed, Osherson, Stob and Weinstein [19] proved that any *non-recursive* IIM M may be replaced by a *non-recursive* set-driven IIM $\hat{M}$ learning at least as much as M does. With the next definition we consider a natural weakening of Definition 2.

Definition 3. **(Schäfer-Richter [20], Osherson et al. [19])** *Let $\mathcal{L}$ be an indexed family.* ***An IIM M is said to be rearrangement-independent*** *with respect to $\mathcal{L}$ iff its output depends only on the range and on the length of its input; that is, iff $M(t_x) = M(\hat{t}_x)$ for all $x \in \mathbb{N}$, all texts t, $\hat{t} \in text(\mathcal{L})$ provided $t_x^+ = \hat{t}_x^+$.*

Whenever the relevant indexed family $\mathcal{L}$ is clear from the context we refer to rearrangement-independent with respect to $\mathcal{L}$ as rearrangement-independent for short. Furthermore, we make the following convention. For all the learning models in this paper we use the prefix s-, and r- to denote the learning model restricted to set-driven and rearrangement-independent IIMs, respectively. For example, *s-LIM* denotes the collection of all indexed families that are *LIM*–inferable by some set-driven IIM. Next we formalize the other inference models that we have mentioned in the introduction.

Definition 4. **(Gold [6])** *Let $\mathcal{L}$ be an indexed family, let L be a language, and let $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ be a hypothesis space.* ***An IIM M CFIN–identifies L from text*** *iff for every text t for L, there exists a $j \in \mathbb{N}$ such that M, when successively fed t, outputs the single hypothesis j, $L = L(G_j)$, and stops thereafter.*

Furthermore, M CFIN–identifies $\mathcal{L}$ with respect to $\mathcal{G}$ iff, for each $L \in range(\mathcal{L})$, M CFIN–identifies L from text with respect to $\mathcal{G}$.

The resulting learning type is denoted by CFIN.

Consequently, every hypothesis produced by a finitely working IIM has to be a correct guess.

The next definition formalizes the different notions of monotonicity.

Definition 5. **(Jantke [8], Wiehagen [24])** *Let L be a language, and let $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ be a hypothesis space.* ***An IIM M is said to identify the language L from text with respect to $\mathcal{G}$***

(A) ***strong-monotonically***

(B) ***monotonically***

(C) ***weak-monotonically***

iff

M CLIM–identifies L from text with respect to $\mathcal{G}$ and for any text $t \in text(L)$ as well as for any two consecutive hypotheses j_x, j_{x+k} which M has produced when fed t_x and t_{x+k} where $k \in \mathbb{N}^+$ the following conditions are satisfied:

(A) $L(G_{j_x}) \subseteq L(G_{j_{x+k}})$

(B) $L(G_{j_x}) \cap L \subseteq L(G_{j_{x+k}}) \cap L$

(C) *if* $t_{x+k}^+ \subseteq L(G_{j_x})$ *then* $L(G_{j_x}) \subseteq L(G_{j_{x+k}})$.

By *CSMON* *CMON*, and *CWMON*, we denote the collection of all indexed families $\mathcal{L}$ for which there are an IIM M and a hypothesis space $\mathcal{G}$ such that M infers $\mathcal{L}$ strong-monotonically, monotonically, and weak–monotonically, respectively, with respect to the hypothesis space $\mathcal{G}$.

Note that the learning types $\lambda SMON$, λMON, and $\lambda WMON$ do heavily depend on $\lambda \in \{E, \varepsilon, C\}$ as the following proposition shows (cf. Lange and Zeugmann [12, 14] and Lange, Zeugmann and Kapur [16]).

Proposition 1.

$$
\begin{array}{ccccc}
ELIM & = & LIM & = & CLIM \\
\cup & & \cup & & \cup \\
EWMON & \subset & WMON & \subset & CWMON \\
\cup & & \cup & & \cup \\
EMON & \subset & MON & \subset & CMON \\
\cup & & \cup & & \cup \\
ESMON & \subset & SMON & \subset & CSMON \\
\cup & & \cup & & \cup \\
EFIN & = & FIN & = & CFIN
\end{array}
$$

Next, we define conservative learning. Intuitively speaking, conservative IIMs maintain their actual hypothesis at least as long they have not seen data contradicting it. Hence, whenever a conservative IIM performs a mind change it is because it has perceived clear ***inconsistency*** between its guess and the input.

Definition 6. (Angluin [2]) *Let $\mathcal{L}$ be an indexed family, let L be a language, and let $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ be a hypothesis space.* ***An IIM M CCONSERVATIVE–identifies L from text with respect to $\mathcal{G}$*** *iff*

(1) *M CLIM–identifies L from text with respect to $\mathcal{G}$,*

(2) *for every text t for L the following condition is satisfied:*
if M on input t_x makes the guess j_x and then outputs the hypothesis $j_{x+k} \neq j_x$ at some subsequent step, then $t^+_{x+k} \not\subseteq L(G_{j_x})$.

Finally, M CCONSERVATIVE–identifies $\mathcal{L}$ with respect to $\mathcal{G}$ iff, for each $L \in range(\mathcal{L})$, M CCONSERVATIVE–identifies L from text with respect to $\mathcal{G}$.

The collection of indexed families *CCONSERVATIVE* is defined in an analogous manner as above. The following proposition completely clarifies the relations between conservative and weak-monotonic learning.

Proposition 2. (Lange and Zeugmann [11])

$\lambda WMON = \lambda CONSERVATIVE$ for all $\lambda \in \{C, \varepsilon, E\}$.

Finally, some of the proofs given below use the notion of uniformly recursively generable families of finite sets $(T_j)_{j\in\mathbb{N}}$. Therefore, we present the definition here. A family of finite sets $(T_j)_{j\in\mathbb{N}}$ is said to be ***uniformly recursively generable*** iff there is a total effective procedure which, on every input j, generates all elements of T_j and stops. If the procedure stops without any output, the corresponding set is empty.

3. Learning with Set-driven IIMs

In this section we study the question under what circumstances set-drivenness does restrict the power of the learning models defined above. We start with finite learning. The next theorem in particular states that finite learning is invariant with respect to the specific choice of the hypothesis space. Moreover, for every hypothesis space comprising the target indexed family $\mathcal{L}$ there is a *set-driven* IIM that finitely learns $\mathcal{L}$.

Theorem 1. *EFIN* = *FIN* = *CFIN* = *s-EFIN*.

Proof. *EFIN* = *FIN* = *CFIN* is due to Lange and Zeugmann [12]. It remains to show that *EFIN* $\subseteq$ *s-EFIN*.

Let $\mathcal{L} \in$ *EFIN*. From the characterization theorem for finite learning (cf. Lange and Zeugmann [10]) it follows that there exists a recursively generable family $(T_j)_{j\in\mathbb{N}}$ of finite nonempty sets such that

(1) for all $j \in \mathbb{N}$, $T_j \subseteq L_j$,

(2) for all $j, k \in \mathbb{N}$, if $T_j \subseteq L_k$, then $L_k = L_j$.

Using this recursively generable family $(T_j)_{j\in\mathbb{N}}$ we define an IIM M witnessing $\mathcal{L} \in$ *s-EFIN*. Let $L \in range(\mathcal{L})$, $t \in text(L)$, and $x \in \mathbb{N}$.

IIM *M*: "On input t_x do the following: If $x = 0$ or $x > 0$ and M, when successively fed t_{x-1}, does not stop, then execute Stage x.

Stage x**:** Search for the least j such that $t_x^+ \subseteq L_j$. Test whether or not $T_j \subseteq t_x^+$.
In case it is, output j and stop.
Otherwise, output nothing and request the next input."

It remains to show that M *s-EFIN*–infers $\mathcal{L}$. By construction, M uses $\mathcal{L}$ as its hypothesis space.

Claim 1. M finitely infers $\mathcal{L}$.

Let $L \in range(\mathcal{L})$ and let $t \in text(L)$. We have to show that M stops sometimes, say with j, and that $L = L_j$. Let k be the least number satisfying $L = L_k$. By Property (1), $T_k \subseteq L_k$. Since $range(t) = L$, there must be an x such that $T_k \subseteq t_x^+$. Now, there is only one case imaginable that could prevent M stopping. Namely, there exists a $j < k$ with $L \subset L_j$ and $T_j \not\subseteq L$. Clearly, in this case M would verify $t_y^+ \subseteq L_j$ but it never verifies $T_j \subseteq t_y^+$. However, since $L \subset L_j$, and $L = L_k$ we conclude $T_k \subseteq L_j$. Therefore, $L = L_j$ by Property (2), a contradiction. Consequently, M has to stop sometimes. Suppose, M, when fed t_y, outputs j and stops. But then, in accordance with M's definition, M has verified $t_y^+ \subseteq L_j$ and $T_j \subseteq t_y^+$. Hence, $T_j \subseteq L_k$. By Property (2), we directly obtain $L_j = L_k = L$. This proves Claim 1.

Claim 2. M is set-driven.

Let $L, \hat{L} \in \mathit{range}(\mathcal{L})$, let $t \in \mathit{text}(L)$, $\hat{t} \in \mathit{text}(\hat{L})$, and let $x, y \in \mathbb{N}$ be any numbers such that $t_x^+ = \hat{t}_y^+$. We have to show that $M(t_x) = M(\hat{t}_y)$. Suppose, M executes Stage x for t_x and Stage y for $\hat{t}_y$, respectively. Since $L \in \mathit{range}(\mathcal{L})$, there exist numbers k, m such that $L = L_k$ and $\hat{L} = L_m$, respectively. Hence, M finds least indices i, j for t_x and $\hat{t}_y$, respectively, such that $t_x^+ \subseteq L_i$ and $\hat{t}_y^+ \subseteq L_j$. Because of $t_x^+ = \hat{t}_y^+$, we may conclude $i = j$. Since the tell-tale sets T_j are uniformly recursively generable, M can effectively compute T_j. If $T_j \not\subseteq t_x^+$, then T_j is not a subset of $\hat{t}_y^+$ either. Hence, in this case M does not output a hypothesis when fed t_x and $\hat{t}_y$, respectively. On the other hand, if $T_j \subseteq t_x^+$ then $T_j \subseteq \hat{t}_y^+$. Therefore, $M(t_x) = M(\hat{t}_y) = j$.

Finally, suppose M has stopped when successively fed t_{x-1}. Clearly, then M has output a hypothesis, say j. We have to show that $M(\hat{t}_y) = j$. Since M finitely infers $\mathcal{L}$, we know that $L_j = L$. Moreover, M has verified that $T_j \subseteq t_z^+ \subseteq L_j$ for some $z < x$. By assumption, $t_x^+ = \hat{t}_y^+$, and therefore $t_z^+ \subseteq \hat{t}_y^+$. We distinguish the following two cases.

Case 1. M when successively fed $\hat{t}_{y-1}$ does not stop.

In accordance with M's definition we know that M executes Stage y on input $\hat{t}_y$. Since j is the least index with $T_j \subseteq t_z^+ \subseteq L_j$ and since $L_j = L$, we conclude $t_x^+ \subseteq L_j$. Because of $t_x^+ = \hat{t}_y^+$ and $T_j \subseteq t_z^+ \subseteq \hat{t}_y^+$, it immediately follows that $T_j \subseteq \hat{t}_y^+ \subseteq L_j$. Hence, M, when successively fed $\hat{t}_y$, has to output a hypothesis, say n, and to stop. Suppose, $n \neq j$. Since $T_j \subseteq \hat{t}_y^+ \subseteq L_j$, we directly obtain $n < j$. On the other hand, $t_z^+ \subseteq \hat{t}_y^+$ and $\hat{t}_y^+ \subseteq L_n$ imply $t_z^+ \subseteq L_n$. Therefore, if $T_n \not\subseteq t_r^+$ for all $r < x$ then M does not stop when successively fed t_{x-1}. However, by assumption M has stopped. Thus it must have verified $T_n \subseteq t_z^+$, since $z < x$. Consequently, $M(t_z) = n$, a contradiction to $n \neq j$. Hence, M, when executing Stage y, outputs j and stops.

Case 2. M when successively fed $\hat{t}_{y-1}$ stops.

By construction, M, when successively fed $\hat{t}_{y-1}$, executes some Stage r, $r \leq y-1$, outputs a hypothesis, say n, and stops. Hence, M has verified $T_n \subseteq \hat{t}_r^+ \subseteq L_n$. We have to show that $n = j$. Suppose the converse, i.e., $j \neq n$. Again, since j is the least index with $T_j \subseteq t_z^+ \subseteq L_j$ and since $L_j = L$ as well as $L_j \supseteq t_x^+ = \hat{t}_y^+$, we may conclude $\hat{t}_r^+ \subseteq L_j$. Therefore, $j \neq n$ implies $n < j$. On the other hand, $n = M(\hat{t}_r)$ implies $L_n = \hat{L}$, since M finitely learns $\mathcal{L}$. Because of $L_n \supseteq \hat{t}_y^+ = t_x^+$, it immediately follows that $t_z^+ \subseteq L_n$. Finally, since $M(t_z) = j$, the definition of M directly implies $j < n$, a contradiction. Hence, we obtain $j = n$. □

As we have already mentioned, the examples of Schäfer-Richter [20] and Fulk [5] witnessing the restriction of set-driven learners are not indexed families. Hence, we ask whether the uniform recursiveness of all target languages may compensate the impact to learn with set-driven IIMs. The answer is no as the following theorem impressively shows.

Theorem 2. *s-CLIM* $\subset$ *ELIM* $=$ *LIM* $=$ *CLIM*.

Proof. The part *ELIM* $=$ *LIM* $=$ *CLIM* is due to Lange and Zeugmann [14]. It remains to show that *s-CLIM* $\subset$ *ELIM*.

The desired indexed family $\mathcal{L}$ is defined as follows. For all $k \in \mathbb{N}$ we set $L_{\langle k,0\rangle} = \{a^k b^n \mid n \in \mathbb{N}^+\}$. For all $k \in \mathbb{N}$ and all $j \in \mathbb{N}^+$ we distinguish the following cases:

Case 1. $\neg\ \Phi_k(k) \leq j$

Then we set $L_{\langle k,j\rangle} = L_{\langle k,0\rangle}$.

Case 2. $\Phi_k(k) \leq j$

Let $d = 2 \cdot \Phi_k(k) - j$. Now, we set:

$$L_{\langle k,j\rangle} = \begin{cases} \{a^k b^m \mid 1 \leq m \leq d\}, & \text{if } d \geq 1, \\ \{a^k b\}, & \text{otherwise.} \end{cases}$$

$\mathcal{L} = (L_{\langle k,j\rangle})_{j,k\in\mathbb{N}}$ is an indexed family of recursive languages, since the predicate '$\Phi_i(y) \leq z$' is uniformly decidable in i, y, and z.

Claim 1. $\mathcal{L} \notin$ *s-CLIM*.

Since the halting problem is undecidable, Claim 1 follows by contraposition of the following Claim 2.

Claim 2. If there exists an IIM M witnessing $\mathcal{L} \in s\text{-}CLIM$, then one can effectively construct an algorithm deciding for all $k \in \mathbb{N}$ whether or not $\varphi_k(k)$ converges.

Let M be any IIM that learns $\mathcal{L}$ in the limit with respect to some hypothesis space $\mathcal{G}$ comprising $range(\mathcal{L})$. We define an algorithm $\mathcal{A}$ that solves the halting problem.

Algorithm $\mathcal{A}$: "On input k execute (A1) and (A2).

(A1) For $z = 0, 1, 2, ...$ generate successively the canonical text t of $L_{\langle k,0\rangle}$ until M on input t_z outputs for the first time a hypothesis j such that $t_z^+ \cup \{a^k b^{z+2}\} \subseteq L(G_j)$.

(A2) Test whether $\Phi_k(k) \le z + 1$. In case it is, output '$\varphi_k(k)$ converges.' Otherwise output '$\varphi_k(k)$ diverges' and stop.

Since M has to infer $L_{\langle k,0\rangle}$ in particular from t, there has to be a least z such that M on input t_z computes a hypothesis j satisfying $t_z^+ \cup \{a^k b^{z+2}\} \subseteq L(G_j)$. Moreover, the test whether or not $t_z^+ \cup \{a^k b^{z+2}\} \subseteq L(G_j)$ can be effectively performed, since membership in $L(G_j)$ is uniformly decidable. By the definition of a complexity measure, Instruction (A2) is effectively executable. Hence, $\mathcal{A}$ is an algorithm.

It remains to show that $\varphi_k(k)$ diverges, if $\neg\, \Phi_k(k) \le z + 1$. Suppose the converse; then there exists a $y > z + 1$ with $\Phi_k(k) = y$. In accordance with the definition of $\mathcal{L}$, we obtain $L = t_z^+ \in \mathcal{L}$. Hence, t_z is also an initial segment of a text $\hat{t}$ for L. Due to the definition of $\mathcal{A}$, we have $L(G_j) \neq L$. Since M is a set-driven IIM, $L = t_z^+$ implies $M(\hat{t}_{z+r}) = j$ for all $r \in \mathbb{N}$. Therefore, M fails to infer L from its text $\hat{t}$. This contradicts our assumption that M is a set-driven IIM which *CLIM*–infers $\mathcal{L}$ with respect to $\mathcal{G}$. Hence, Claim 2 is proved.

Claim 3. $\mathcal{L} \in ELIM$.

After a little reflection, it is easy to verify that the following IIM M infers $\mathcal{L}$ in the limit with respect to the hypothesis space $\mathcal{L}$. Let $L \in range(\mathcal{L})$, let $t \in text(L)$, and let $x \in \mathbb{N}$. We define:

IIM M: "On input t_x do the following: Determine the unique k such that $t_0 = a^k b^m$ for some $m \in \mathbb{N}$. Test whether or not $\Phi_k(k) \le x$. In case it is, goto (1). Otherwise, output $\langle k, 0\rangle$ and request the next input.

(1) Test whether or not $a^k b^{\Phi_k(k)+n} \in t_x^+$ for some $n \in \mathbb{N}$. In case it is, output $\langle k, 0\rangle$ and request the next input. Otherwise, goto (2).

(2) Determine the maximal $z \in \mathbb{N}$ such that $a^k b^z \in t_x^+$. Output $\langle k, 2 \cdot \Phi_k(k) - z\rangle$ and request the next input."

□

As the latter theorem shows, sometimes there is no way to design a set-driven IIM. However, with the following theorems we mainly intend to show that the careful choice of the hypothesis space deserves special attention whenever set-drivenness is desired. The first result nicely contrasts the fact that unconstraint language learning is invariant with respect to the particular choice of the hypothesis space.

Theorem 3. *s-ELIM* $\subset$ *s-LIM* $\subset$ *s-CLIM*.

Proof. By definition *s-ELIM* $\subseteq$ *s-LIM* $\subseteq$ *s-CLIM*. It remains to show that the stated inclusions are proper. First, we separate *s-LIM* and *s-ELIM*.

The target indexed family $\mathcal{L} = (L_{\langle k,j\rangle})_{k,j\in\mathbb{N}}$ is defined as follows. Without loss of generality, we may assume that $\Phi_k(k) \geq 1$ for all $k \in \mathbb{N}$. We set $L_{\langle k,0\rangle} = \{a^k b\} \cup \{a^k b^{\Phi_k(k)+1}\}$ for all $k \in \mathbb{N}$. Notice that $\{a^k b^{\Phi_k(k)+1}\}$ equals the empty set, if $\varphi_k(k)$ is undefined. Furthermore, for all $j \geq 1$ we distinguish the following cases.

Case 1. $\neg\ \Phi_k(k) = j$

Then let $L_{\langle k,j\rangle} = \{a^k b, a^k b^{j+1}\}$.

Case 2. $\Phi_k(k) = j$

Then we set $L_{\langle k,j\rangle} = \{a^k b\}$.

Obviously, $\mathcal{L} = (L_{\langle k,j\rangle})_{k,j\in\mathbb{N}}$ constitutes an indexed family.

Claim 1. $\mathcal{L} \in$ *s-LIM*.

Consider the following hypothesis space $\mathcal{G} = (G_{\langle k,j\rangle})_{k,j\in\mathbb{N}}$. For all $k,\ j \in \mathbb{N}$ let $L(G_{\langle k,j\rangle}) = \{a^k b, a^k b^{j+1}\}$. After a little reflection, it is not hard to see that $range(\mathcal{L}(\mathcal{G})) = range(\mathcal{L})$. Now, a set-driven IIM M which *LIM*–infers $\mathcal{L}$ with respect to $\mathcal{G}$ can be designed as follows. Given any initial segment t_x of any text t for any language $L \in range(\mathcal{L})$, M simply determines the relevant k and searches the least index j such that $t_x^+ \subseteq G_{\langle k,j\rangle}$. Then it outputs the hypothesis $\langle k,j\rangle$. We omit the details.

Claim 2. $\mathcal{L} \notin s\text{-}ELIM$.

This claim is proved via the following lemma.

Lemma 1. *Let M be any set-driven IIM witnessing $\mathcal{L} \in ELIM$. Then M may be used to decide the halting problem.*

Proof. We define an algorithm $\mathcal{B}$ as follows.

Algorithm $\mathcal{B}$: "On input k execute Instruction (B).

(B) Compute $\langle k,j\rangle = M(a^k b)$. If $j = 0$, output '$\varphi_k(k)\uparrow$' and stop.
Else, output '$\varphi_k(k)\downarrow$,' and stop."

By assumption, M has to learn $\{a^k b\}$ in the limit from its unique text $t = a^k b,\ a^k b, \ldots$. Since M is set-driven, it must make an output on input $a^k b$. Moreover, this output has to be a correct hypothesis, i.e., $M(a^k b) = \langle k,j\rangle$, and $\{a^k b\} = L_{\langle k,j\rangle}$. Therefore, $\mathcal{B}$ terminates on every input k. It remains to show that $\mathcal{B}$ behaves correctly.

Suppose, $\mathcal{B}$ outputs '$\varphi_k(k)\uparrow$' but '$\varphi_k(k)\downarrow$.' Let $\Phi_k(k) = y$. By assumption, $y \geq 1$. Now, looking at $\mathcal{L}$'s definition, one easily verifies that $L_{\langle k,0\rangle} = \{a^k b, a^k b^{y+1}\}$. But this directly contradicts $L_{\langle k,0\rangle} = \{a^k b\}$. Hence, $\mathcal{B}$ behaves correctly if it outputs '$\varphi_k(k)\uparrow$.'

Now suppose that $\mathcal{B}$ outputs '$\varphi_k(k)\downarrow$.' By construction, $j \neq 0$, and as already mentioned $L_{\langle k,j\rangle} = \{a^k b\}$. Taking $\mathcal{L}$'s definition into account we can conclude that $j = \Phi_k(k)$. Consequently, $\varphi_k(k)$ is indeed defined. This proves Lemma 1.

Since the halting problem is not recursive, the contraposition of Lemma 1 yields Claim 2.

Finally, we have to separate *s-CLIM* and *s-LIM*. However, this separation is an immediate consequence of the following slightly stronger theorem, and therefore we omit its proof here. □

Theorem 4. *There is an indexed family $\mathcal{L}$ such that*

(1) $\mathcal{L} \in r\text{-}ESMON$,

(2) $\mathcal{L} \notin$ *s-LIM*,

(3) $\mathcal{L} \in$ *s-CSMON*.

Proof. The desired indexed family $\mathcal{L} = (L_{\langle k,j\rangle})_{k,j\in\mathbb{N}}$ is defined as follows. For all $k \in \mathbb{N}$ we set $L_{\langle k,0\rangle} = \{a^k b^m \,|\; m \in \mathbb{N}^+\}$. For all $j \in \mathbb{N}^+$ we distinguish the following cases.

Case 1. $\neg\ \Phi_k(k) \leq j$

Then we define $L_{\langle k,j\rangle} = L_{\langle k,0\rangle}$.

Case 2. $\Phi_{k_1}(k_1) \leq j$

Then we set $L_{\langle k,j\rangle} = \{a^k b\}$.

Claim 1. $\mathcal{L} \in$ *r-ESMON*.

We have to define an IIM witnessing $\mathcal{L} \in$ *r-ESMON*. This is done as follows: Let $L \in range(\mathcal{L})$, $t \in text(L)$, and $x \in \mathbb{N}$.

IIM *M*: "On input t_x do the following: Compute the unique k such that $a^k b^m \in t_x^+$ for some $m \in \mathbb{N}$. As long as $t_x^+ = \{a^k b\}$ execute (A).
Otherwise, output $\langle k, 0\rangle$ and request the next input.

(A) Test whether or not $\neg\ \Phi_k(k) \leq x$. In case it is, output nothing and request the next input.
Otherwise, output $\langle k, \Phi_k(k)\rangle$ and request the next input."

Obviously, M is rearrangement-independent. In order to prove that M *ESMON*–infers $\mathcal{L}$ we distinguish the following cases.

Case 1. $\varphi_k(k)\uparrow$

In accordance with the definition of the indexed family $\mathcal{L}$ we directly obtain $L = L_{\langle k,0\rangle} = L_{\langle k,j\rangle}$ for all $j \in \mathbb{N}$. Since $t \in text(L)$, there has to be an x such that $t_x^+ \neq \{a^k b\}$. Consequently, after having seen t_x the IIM M always outputs $\langle k, 0\rangle$, a correct hypothesis. Moreover, M obviously behaves strong-monotonically.

Case 2. $\varphi_k(k)\downarrow$

Suppose, $L = \{a^k b\}$. Since $t \in \textit{text}(L)$, M executes Instruction (A) on every input t_x. Moreover, there exists an x_0 such that $\Phi_k(k) \leq x$ for all $x \geq x_0$. Hence, after having seen t_{x_0} the IIM M always outputs the correct hypothesis $\langle k, \Phi_k(k)\rangle$.

Now, let us assume $L = L_{\langle k,0\rangle}$. As we have shown in Case 1, there exists an $x \in \mathbb{N}$ such that $t_x^+ \neq \{a^k b\}$. Consequently, for all $y \geq x$ we have $M(t_y) = \langle k, 0\rangle$ and M again learns L. Finally, it might happen that M outputs $\langle k, \Phi_k(k)\rangle$ on some initial segment of t and changes its mind to $\langle k, 0\rangle$ afterward. Clearly, this mind change fulfills the strong-monotonicity constraint. Hence, M witnesses $\mathcal{L} \in r\text{-}\mathit{ESMON}$.

Claim 2. $\mathcal{L} \notin s\text{-}\mathit{LIM}$.

Applying *mutatis mutandis* the same idea underlying the proof of Theorem 3, Claim 2, $\mathcal{L} \notin s\text{-}\mathit{LIM}$ can be shown by reducing the halting problem to $\mathcal{L} \in s\text{-}\mathit{LIM}$. For the sake of completeness we present the modification of Algorithm $\mathcal{B}$.

Algorithm $\tilde{\mathcal{B}}$: "On input k execute Instruction (B1).

(B1) Simulate M on input $a^k b$. If M requests the next input without outputting a hypothesis, then output '$\varphi_k(k) \uparrow$,' and stop.
Otherwise, let $z = M(a^k b)$. Execute Instruction (B2).

(B2) Test whether or not $a^k b^2 \in L(G_z)$.
In case it is, output '$\varphi_k(k) \uparrow$.'
Else, output '$\varphi_k(k) \downarrow$,' and stop."

Claim 3. $\mathcal{L} \in s\text{-}\mathit{CSMON}$.

First of all we define the desired class-comprising hypothesis space. For all $j \in \mathbb{N}$ we set

$$L(G_j) = \begin{cases} L_{\langle k,0\rangle}, & \text{if } j = 2k, \\ \{a^k b\}, & \text{if } j = 2k+1. \end{cases}$$

Obviously, $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ is an admissable hypothesis space. A set-driven IIM M witnessing $\mathcal{L} \in \mathit{CSMON}$ may be easily defined as follows. As long as it receives an initial segment t_x of a text t such that $t_x^+ = \{a^k b\}$ it outputs $2k+1$. If $\{a^k b\} \subset t_x^+$, it hypothesizes $2k$. We omit the details. □

Theorem 4 directly yields the following corollary which relates the power of set-driven and unrestricted IIMs to one another.

Corollary 5. *For all* $ID \in \{SMON, MON, WMON\}$*:*

(1) $s\text{-}EID \subset EID$,

(2) $s\text{-}ID \subset ID$.

Proof. By Theorem 4 we have $r\text{-}ESMON \setminus s\text{-}LIM \neq \emptyset$. This yields all the proper inclusions mentioned, since $ESMON \subset SMON$, $ESMON \subset EID \subset ID$ for all $ID \in \{MON, WMON\}$ (cf. Proposition 1) as well as $s\text{-}EID \subseteq s\text{-}ID \subseteq s\text{-}LIM$ for all $ID \in \{SMON, MON, WMON\}$ by definition. □

As we have seen, set-drivenness constitutes a severe restriction. While this is true in general as long as exact and class-preserving learning is considered, the situation is different in the class-comprising case. On the one hand, learning in the limit cannot always be achieved by set-driven IIMs (cf. Theorem 2). On the other hand, conservative learners may always be designed to be set-driven, if the hypothesis space is appropriately chosen.

Theorem 6. $s\text{-}CCONSERVATIVE = CCONSERVATIVE$.

Proof. We partition the proof into two parts. First, we show that every indexed family in *CCONSERVATIVE* belongs to *r-CCONSERVATIVE* (cf. Lemma 2) below. Then we apply this result and show that set-drivenness does not restrict the power of class-comprising conservative learning (cf. Lemma 3).

Lemma 2. $r\text{-}CCONSERVATIVE = CCONSERVATIVE$.

Let $\mathcal{L} \in CCONSERVATIVE$. By Theorem 14 in Lange and Zeugmann [13] there exist a hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ and a recursively generable tell-tale family $(T_j)_{j\in\mathbb{N}}$ of finite and nonempty sets such that

(1) $range(\mathcal{L}) \subseteq range(\mathcal{L}(\mathcal{G}))$,

(2) for all $j \in \mathbb{N}$, $T_j \subseteq L(G_j)$,

(3) for all $j, k \in \mathbb{N}$, if $T_j \subseteq L(G_k)$, then $L(G_k) \not\subset L(G_j)$.

Using this tell-tale family, we define a new recursively generable family $(\hat{T}_j)_{j\in\mathbb{N}}$ of finite and nonempty sets that allows the design of a rearrangement-independent IIM inferring $\mathcal{L}$ conservatively with respect to $\mathcal{G}$. For all $j \in \mathbb{N}$ we set $\hat{T}_j = \bigcup_{n\leq j} T_n \cap L(G_j)$.

It is easy to see that $(\hat{T}_j)_{j\in\mathbb{N}}$ fulfills (2) and (3), too. Next, we show the following even stronger result.

Statement 1. $\mathcal{L}(\mathcal{G}) \in$ *r-ECONSERVATIVE.*

The desired IIM is defined as follows. Let $L \in range(\mathcal{L}(\mathcal{G}))$, $t \in text(L)$, and $x \in \mathbb{N}$.

IIM *M*: "On input t_x do the following: Generate $\hat{T}_k$ for all $k \leq x$ and test whether $\hat{T}_k \subseteq t_x^+ \subseteq L(G_k)$. In case there is one k fulfilling the test, output the minimal one, and request the next input.

Otherwise, output nothing and request the next input."

Obviously, M is rearrangement-independent.

Claim 1. M is conservative.

Let k and j be two distinct hypotheses produced by M on input t_x and t_{x+r}, respectively. We have to show that $t_{x+r}^+ \not\subseteq L(G_k)$. For that purpose we distinguish the following cases.

Case 1. $k < j$

Due to M's definition we immediately obtain $t_{x+r}^+ \not\subseteq L(G_k)$.

Case 2. $j < k$

Suppose, $t_{x+r}^+ \subseteq L(G_k)$. In accordance with its definition, M has verified that $\hat{T}_j \subseteq t_{x+r}^+ \subseteq L(G_j)$. Moreover, the definition of the tell-tale family directly yields $\hat{T}_j \subseteq \hat{T}_k$, since $j < k$ and $\hat{T}_j \subseteq t_{x+r}^+ \subseteq L(G_k)$. Taking into account that $\hat{T}_k \subseteq t_x^+$, this implies $\hat{T}_j \subseteq t_x^+ \subseteq L(G_j)$. Finally, since $j < k$ we conclude $M(t_x) = j$, a contradiction. Hence, Claim 1 is proved.

Claim 2. M infers L from t.

Let z be the least k such that $L(G_k) = L$, hereafter denoted as $z = \mu k\ [L(G_k) = L]$. Therefore, $L(G_j) \neq L$ for all $j < z$. Applying Property (3), we obtain that $L \setminus L(G_j) \neq \emptyset$ for all $j < z$ provided $\hat{T}_j \subseteq L$. Consequently, every candidate hypothesis $j < z$ is sometimes rejected by M, and M converges to z. Hence, Claim 2 follows and Statement 1 is proved.

Finally, since $range(\mathcal{L}) \subseteq range(\mathcal{L}(\mathcal{G}))$, we may conclude that M *r-CCONSERVATIVE*–infers $\mathcal{L}$. This completes the proof of Lemma 2.

Lemma 3. *Let $\mathcal{L}$ be any indexed family. If $\mathcal{L} \in$ CCONSERVATIVE, then there exist a hypothesis space $\tilde{\mathcal{G}} = (\tilde{G}_j)_{j\in\mathbb{N}}$ and an IIM $\tilde{M}$ witnessing $\mathcal{L} \in$ s-CCONSERVATIVE with respect to $\tilde{\mathcal{G}}$.*

First, we define the hypothesis space $\tilde{\mathcal{G}} = (\tilde{G}_j)_{j\in\mathbb{N}}$ as follows. Applying Lemma 2, there are a hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ and an IIM M such that M *r-CCONSERVATIVE*–learns $\mathcal{L}$ with respect to $\mathcal{G}$. Moreover, M witnesses $\mathcal{L}(\mathcal{G}) \in$ *r-ECONSERVATIVE*. Afterwards, we use the latter statement and show a more general result which turns out to be quite helpful in order to prove Corollary 7. The hypothesis space $\tilde{\mathcal{G}}$ is the canonical enumeration of all grammars from $\mathcal{G}$ and all finite languages over the underlying alphabet Σ. Second, the main ingredient to the definition of the desired IIM $\tilde{M}$ is the machine M from Lemma 2. However, before defining it we introduce the notion of *repetition free* text $rf(t)$. Let $t = s_0, s_1, ...$ be any text. We set $rf(t_0) = s_0$ and proceed inductively as follows: For all $x \geq 1$, $rf(t_{x+1}) = rf(t_x)$, if $s_{x+1} \in rf(t_x)^+$, and $rf(t_{x+1}) = rf(t_x) \cdot s_{x+1}$ otherwise. Obviously, given any initial segment t_x of a text t one can effectively compute $rf(t_x)$.

Statement 2. $\mathcal{L}(\mathcal{G}) \in$ *s-CCONSERVATIVE* with respect to $\tilde{\mathcal{G}}$.

Now, let $L \in range(\mathcal{L}(\mathcal{G}))$, $t \in text(L)$, and $x \in \mathbb{N}$.

IIM $\tilde{M}$: "On input t_x do the following: Compute $rf(t_x)$. If M on input $rf(t_x)$ outputs a hypothesis, say j, then output the canonical index of j in $\tilde{\mathcal{G}}$ and request the next input. Otherwise, output the canonical index of t_x^+ in $\tilde{\mathcal{G}}$ and request the next input."

Claim 3. $\tilde{M}$ is set-driven.

Let t, $\hat{t} \in text(\mathcal{L})$, and let $x, y \in \mathbb{N}$ such that $t_x^+ = \hat{t}_y^+$. We have to show that $\tilde{M}(t_x) = \tilde{M}(\hat{t}_y)$. Clearly, $length(rf(t_x)) = length(rf(\hat{t}_y))$, and therefore we conclude $M(rf(t_x)) = M(rf(\hat{t}_y))$, since M is rearrangement-independent. That means, either M outputs in both cases the same hypothesis or it outputs nothing on input $rf(t_x)$ and $rf(\hat{t}_y)$, respectively. This proves Claim 3.

Claim 4. $\tilde{M}$ is conservative.

By construction, $\tilde{M}$ outputs on every input a hypothesis. Let $j = \tilde{M}(t_x)$ and $k = \tilde{M}(t_{x+1})$ with $j \neq k$. Since $\tilde{M}$ is set-driven, we obtain $t_x^+ \subset t_{x+1}^+$. We consider the following cases.

Case 1. M on input $rf(t_x)$ does not output a hypothesis.

Then $L(\tilde{G}_j) = t_x^+$, and consequently, $t_{x+1}^+ \not\subseteq L(\tilde{G}_j)$. Hence, $\tilde{M}$ performs a justified mind change.

Case 2. M on input $rf(t_x)$ outputs a hypothesis.

Obviously, $rf(t_x)$ is a proper initial segment of $rf(t_{x+1})$. Suppose, M on input $rf(t_{x+1})$ produces a hypothesis, too. Since M is conservative, we immediately obtain $t_{x+1} \not\subseteq L(\tilde{G}_j)$. Hence, it remains to consider the scenario in which M on input $rf(t_{x+1})$ does not produce a hypothesis. Looking at M's definition, we see that M could be prevented from doing it only by detecting an inconsistency. Consequently, $\tilde{M}$ is conservative, and Claim 4 is proved.

Claim 5. $\tilde{M}$ infers L from t.

Again, we distinguish two cases.

Case 1. L is finite.

Then there exists an $x \in \mathbb{N}$ such that $t_x^+ = L$. Moreover, if M on input $rf(t_x)$ produces a hypothesis, then it is a correct one, since M is conservative. Hence, in this case $\tilde{M}$ infers L from t. On the other hand, if M on input $rf(t_x)$ does not output a hypothesis, then $\tilde{M}$ converges to the canonical index of the finite language t_x^+ in $\tilde{\mathcal{G}}$, since $\tilde{M}$ is set-driven.

Case 2. L is infinite.

Since L is infinite, $rf(t)$ is a text for L, too. Moreover, M has to infer L in particular from $rf(t)$. Therefore, there exists an $x \in \mathbb{N}$ such that $M(rf(t)_{x+r}) = k$ with $L(G_k) = L$ for all $r \in \mathbb{N}$. Hence, after some point $\tilde{M}$ exclusively outputs the canonical index of $L(G_k)$ in $\tilde{\mathcal{G}}$. Consequently, $\tilde{M}$ infers L. This completes the proof of Claim 5.

Therefore, Statement 2 is proved. Since $range(\mathcal{L}) \subseteq range(\mathcal{L}(\mathcal{G}))$, we immediately may conclude that $\tilde{M}$ *CCONSERVATIVE*–learns $\mathcal{L}$ with respect to $\tilde{\mathcal{G}}$. This completes the proof of Lemma 3. □

Corollary 7. *Let $\mathcal{L} \in$ CCONSERVATIVE. Then, there exists a hypothesis space*

$\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ *comprising* $range(\mathcal{L})$ *such that* $L(\hat{\mathcal{G}}) \in$ *s-ECONSERVATIVE*.

Proof. Let $\mathcal{L} \in$ *CCONSERVATIVE*. Furthermore, due to the latter theorem, there are a set-driven IIM $\tilde{M}$ and a hypothesis space $\tilde{\mathcal{G}}$ such that $\tilde{M}$ conservatively infers $\mathcal{L}$ with respect to $\tilde{\mathcal{G}}$. Let $\tilde{M}$ and $\tilde{\mathcal{G}}$ be defined as in Lemma 3.

Recall that $\tilde{\mathcal{G}}$ is a canonical enumeration of $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ with $range(\mathcal{L}) \subseteq range(\mathcal{L}(\mathcal{G}))$ and of all finite languages over the underlying alphabet. Without loss of generality we may assume that $\tilde{\mathcal{G}}$ fulfills the following property. If j is even, then $L(\tilde{G}_j) \in range(\mathcal{L}(\mathcal{G}))$. Hence, $\tilde{M}$ *s-CCONSERVATIVE*–learns $L(\tilde{G}_j)$ with respect to $\tilde{\mathcal{G}}$. Otherwise, $L(\tilde{G}_j)$ is a finite language.

We start with the definition of the desired hypothesis space $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ If j is even, then we set $\hat{G}_j = \tilde{G}_j$. Otherwise, we distinguish the following cases. If M, when fed the lexicographically ordered enumeration of all strings in $L(\tilde{G}_j)$ outputs the hypothesis j, then we set $\hat{G}_j = \tilde{G}_j$. In case it does not, we set $\hat{G}_j = \tilde{G}_{j-1}$.

Now we are ready to define the desired IIM M witnessing $L(\hat{\mathcal{G}}) \in$ *s-ECONSERVATIVE*. Let $L \in range(\mathcal{L}(\hat{\mathcal{G}}))$, $t \in text(L)$, and $x \in \mathbb{N}$.

IIM *M*: "On input t_x do the following: Simulate $\tilde{M}$ on input t_x. If $\tilde{M}$ does not output any hypothesis, then output nothing and request the next input.
Otherwise, let $j = \tilde{M}(t_x)$. Output j and request the next input."

Since $\tilde{M}$ is a conservative and set-driven IIM, M behaves thus. It remains to show that M learns L. Obviously, if $L = L(\hat{G}_{2k})$ for some $k \in \mathbb{N}$, then $\tilde{M}$ infers L, since $\tilde{M}$ *s-CCONSERVATIVE*–infers $\mathcal{L}$. Therefore, since M simulates $\tilde{M}$, we are done.

Now, let us suppose, $L \neq L(\hat{G}_{2k})$ for all $k \in \mathbb{N}$. By definition of $\hat{\mathcal{G}}$, we know that L is finite. Moreover, since t is a text for L, there exists an x such that $t_y^+ = L$ for all $y \geq x$. Recalling the definition of $\hat{\mathcal{G}}$, and by assumption, we obtain the following. There is a number j such that $\tilde{M}(t_x) = j$, $L = t_x^+ = L(\tilde{G}_j) = L(\hat{G}_j)$. Hence, $M(t_x) = j$, too. Finally, since M is set-driven, we directly get $M(t_y) = j$ for all $y \geq x$. Consequently, M learns L. □

Next we characterize class-comprising set-driven learning in the limit.

Theorem 8. *s-CLIM = CCONSERVATIVE.*

Proof. *CCONSERVATIVE* $\subseteq$ *s-CLIM* immediately follows from Theorem 6. It remains to handle the part *s-CLIM* $\subseteq$ *CCONSERVATIVE*.

Let $\mathcal{L}$ be any indexed family and $\hat{M}$ a set-driven IIM which learns $\mathcal{L}$ with respect to a hypothesis space $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$. Without loss of generality, we may assume that $\hat{M}$, when fed any finite sequences σ of strings over the terminal alphabet Σ, always outputs a consistent hypothesis, i.e., if $\hat{M}(\sigma) = j$ then $\sigma^+ \subseteq L(\hat{G}_j)$ (cf. Lange, Zeugmann and Kapur [16] for a detailed discussion).

We show that $\hat{M}$ can be simulated by a conservative IIM M which also learns $\mathcal{L}$. First, we define a suitable hypothesis space $\mathcal{G} = (G_{\langle j,k\rangle})_{j,k\in\mathbb{N}}$.

For the sake of readability, in the following we consider the set-driven IIM $\hat{M}$ as a learning device which receives finite sets of strings as input instead of finite sequences. Let $F_0,\ F_1,\ F_2,\ \dots$ denote any effective repetition-free enumeration of all finite subsets from Σ^*. Given any finite set $F \subseteq \Sigma^*$, we denote by $\#(F)$ the uniquely determined index n with $F_n = F$. For all $j,\ k \in \mathbb{N}$ let us distinguish the following cases.

If $\hat{M}(F_k) \neq j$, then we set $L(G_{\langle j,k\rangle}) = F_k$. Otherwise, i.e., $\hat{M}(F_k) = j$, the language $L(G_{\langle j,k\rangle})$ will be specified as a subset of $L(\hat{G}_j)$ such that $F_k \subseteq L(G_{\langle j,k\rangle})$. For that purpose, we define $L(G_{\langle j,k\rangle})$ via its characteristic function $f_{L(G_{\langle j,k\rangle})}$.

Let $s_0,\ s_1\ s_2,\ \dots$ be the lexicographically ordered enumeration of all strings in Σ^*. Let $m = 0$. If $s_0 \in F_k$, then set $f_{L(G_{\langle j,k\rangle})}(s_0) = 1$. If $s_0 \in L(\hat{G}_j) \setminus F_k$ and $\hat{M}(F_k \cup \{s_0\}) = j$, then let $f_{L(G_{\langle j,k\rangle})}(s_0) = 1$. Otherwise, we set $f_{L(G_{\langle j,k\rangle})}(s_0) = 0$. Now we proceed inductively. Let $m \in \mathbb{N}$ and $F = \{s_n \mid\ n \leq m,\ f_{L(G_{\langle j,k\rangle})}(s_n) = 1\}$. Then we define:

$$f_{L(G_{\langle j,k\rangle})}(s_{m+1}) = \begin{cases} 1, & \text{if } s_{m+1} \in F_k, \\ 1, & \text{if } s_{m+1} \in L(\hat{G}_j) \setminus F_k,\ \hat{M}(F_k \cup V \cup \{s_{m+1}\}) = j \text{ for all } V \subseteq F, \\ 0, & \text{otherwise.} \end{cases}$$

By construction, $\mathcal{G}$ defines a hypothesis space having a uniformly decidable membership problem. First we show that $range(\mathcal{L}) \subseteq range(\mathcal{L}(\mathcal{G}))$. Let $L \in range(\mathcal{L})$. Since $\hat{M}$ is a set-driven IIM that learns L, there has to be a finite set $F \subseteq L$ such that, for all finite sets

$V \subseteq L$, $\hat{M}(F) = \hat{M}(F \cup V) = j$, and $L(\hat{G}_j) = L$ (cf. Fulk [5]). Hence, we may conclude that $L = L(G_{\langle j,k \rangle})$, where $k = \#(F)$. Consequently, $\mathcal{G}$ defines an admissible hypothesis space which comprises $range(\mathcal{L})$.

Furthermore, $\mathcal{G}$'s definition immediately implies:

Observation 1. Let $V \subseteq L(\hat{G}_j)$ and $k \in \mathbb{N}$ such that $\hat{M}(F_k) = j$. Then $V \subseteq L(G_{\langle j,k \rangle})$ implies $\hat{M}(F_k \cup V) = j$.

Now, we are ready to define the desired conservative IIM M which works with respect to the hypothesis space $\mathcal{G}$. Let $L \in range(\mathcal{L})$, $t \in text(L)$, and $x \in \mathbb{N}$.

IIM *M*: "On input t_x do the following: If $x = 0$, compute $\hat{M}(t_0^+) = j$. Output the guess $\langle j, \#(t_0^+) \rangle$ and request the next input. Otherwise, goto (A).

(A) Let $M(t_{x-1}) = \langle j, k \rangle$. Test whether or not $t_x^+ \subseteq L(G_{\langle j,k \rangle})$. In case it is, output $\langle j, k \rangle$ and request the next input.
Else, compute $\hat{M}(t_x^+) = z$. Output the guess $\langle z, \#(t_x^+) \rangle$ and request the next input."

By definition, M outputs in every step a guess. Moreover, M performs exclusively justified mind changes. Thus, M is a conservative IIM. Next we show that M learns L from text t.

Claim 1. When fed t, M never outputs a guess z such that $L \subset L(G_z)$.

Let $x \in \mathbb{N}$, $M(t_x) = z$, and let $y \leq x$ be the least index such that $M(t_y) = z$. By definition $z = \langle j, k \rangle$ where $\hat{M}(t_y^+) = j$ and $k = \#(t_y^+)$. Thus $\hat{M}(t_y^+) = \hat{M}(F_k) = j$, and consequently $L(G_{\langle j,k \rangle}) \subseteq L(\hat{G}_j)$ in accordance with the definition of $\mathcal{G}$. Recall that $t_x^+ \subseteq L(G_{\langle j,k \rangle})$. Hence, $t_x^+ \subseteq L(\hat{G}_j)$, too. Therefore, the assumptions of Observation 1 are fulfilled and we conclude $\hat{M}(t_x^+) = j$. We distinguish the following two cases:

Case 1. $t_x^+ = L$

Therefore, $L(\hat{G}_j) = L$, since $\hat{M}$ is a set-driven IIM which learns L. Because of $t_x^+ \subseteq L(G_{\langle j,k \rangle}) \subseteq L(\hat{G}_j)$, we may conclude $L = L(G_{\langle j,k \rangle})$.

Case 2. $t_x^+ \subset L$

Suppose the converse, that is $L \subset L(G_{\langle j,k \rangle})$. Because of $L(G_{\langle j,k \rangle}) \subseteq L(\hat{G}_j)$, we also have $L \subset L(\hat{G}_j)$. Since $\hat{M}$ learns L, there is an $r \in \mathbb{N}^+$ such that $\hat{M}(t^+_{x+r}) = n$ and $L(\hat{G}_n) = L$. Obviously, $L \subset L(\hat{G}_j)$ implies $t^+_{x+r} \subseteq L(\hat{G}_j)$. On the other hand, $L \subset L(\hat{G}_j)$ yields immediately $j \neq n$. Thus, by contraposition of Observation 1, we may conclude that $t^+_{x+r} \not\subset L(G_{\langle j,k\rangle})$. This contradicts $L \subset L(G_{\langle j,k\rangle})$, since $t^+_{x+r} \subseteq L$.

It remains to show that M, when fed t, converges to a correct guess for L. Since M never outputs an overgeneralized hypothesis and, additionally, M exclusively performs justified mind changes, it suffices to show:

Claim 2. When fed t, M outputs at least once a correct guess.

Since $\hat{M}$ is a set-driven IIM which learns L, there has to be a a finite set $F \subseteq L$ such that, for all finite sets $V \subseteq L$, $\hat{M}(F) = \hat{M}(F \cup V) = j$, and $L(\hat{G}_j) = L$ (cf. Fulk [5]). By $\mathcal{G}$'s definition $L(G_{\langle j,\#(F)\rangle}) = L$ as well as $L(G_{\langle j,\#(F\cup V)\rangle}) = L$ for all finite sets $V \subseteq L$. Since t defines a text for L there has to be a least $x \in \mathbb{N}$ such that $F \subseteq t^+_x$. Let $M(t_x) = z$. If $z = \langle j, \#(t^+_x)\rangle$, we are done. Otherwise, $L \setminus L(G_z) \neq \emptyset$ by Claim 1. Again, since t is a text for L, there has to be a least $r \in \mathbb{N}^+$ such that $t^+_{x+r} \not\subseteq L(G_z)$. By its definition M changes ist mind to the correct guess $\langle j, \#(t^+_{x+r})\rangle$ when processing t_{x+r}.

Hence, M converges to a correct hypothesis for L. To sum up, M conservatively infers $\mathcal{L}$ with respect to the hypothesis space $\mathcal{G}$. □

At this point it is only natural to ask whether or not the latter theorem remains valid if class-comprising learning is replaced by class-preserving inference and exact identification, respectively. The negative answer is provided by our next theorem. Additionally, this theorem gives some more evidence that set-drivenness is not that restrictive as it might seem.

Theorem 9.

(1) *s-SMON* $\setminus$ *EWMON* $\neq \emptyset$,

(2) *s-CSMON* $\setminus$ *WMON* $\neq \emptyset$,

(3) *s-ELIM* $\setminus$ *WMON* $\neq \emptyset$,

(4) *s-EWMON* $\setminus$ *MON* $\neq \emptyset$.

Proof. First of all, we show Assertion (1). Let us consider the following indexed family $\mathcal{L}_{sm} = (L_{\langle k,j\rangle})_{j,k\in\mathbb{N}}$. For all $k \in \mathbb{N}$ we set $L_{\langle k,0\rangle} = \{a^k b^n \mid n \in \mathbb{N}^+\}$. For all $k \in \mathbb{N}$ and all $j \in \mathbb{N}^+$ we distinguish the following cases:

Case 1. $\neg\ \Phi_k(k) \leq j$

We set $L_{\langle k,j\rangle} = L_{\langle k,0\rangle}$.

Case 2. $\Phi_k(k) \leq j$

Then we set $L_{\langle k,j\rangle} = \{a^k b^m \mid 1 \leq m \leq \Phi_k(k)\}$.

In Lange and Zeugmann [12] it was already shown that the family $\mathcal{L}_{sm}$ is witnessing $SMON \setminus EWMON \neq \emptyset$. Hence, it remains to show the following claim.

Claim 1. $\mathcal{L}_{sm} \in s\text{-}SMON$.

We have to show that there are a hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ with $range(\mathcal{L}_{sm}) = range(\mathcal{L}(\mathcal{G}))$ and a set-driven IIM M such that M strong-monotonically infers $\mathcal{L}$ with respect to $\mathcal{G}$.

First of all, we define the hypothesis space $\mathcal{G}$. For all $k \in \mathbb{N}$, we set $L(G_{2k}) = \bigcap_{j\in\mathbb{N}} L_{\langle k,j\rangle}$ and $L(G_{2k+1}) = L_{\langle k,0\rangle}$.

Since $\mathcal{L}_{sm}$ is an indexed family, it is easy to verify that membership is uniformly decidable for $\mathcal{G}$. Moreover, we have $range(\mathcal{L}_{sm}) = range(\mathcal{L}(\mathcal{G}))$.

Let $L \in range(\mathcal{L})_{sm}$, let t be any text for L, and let $x \in \mathbb{N}$. The desired IIM M is defined as follows.

IIM *M*: "On input t_x do the following: Determine the unique k such that $t_0 = a^k b^m$ for some $m \in \mathbb{N}$. Test whether or not $t_x^+ \subseteq L(G_{2k})$. In case it is, output $2k$. Otherwise, output $2k+1$."

Obviously, M changes its mind at most once. Since $L(G_{2k}) \subseteq L(G_{2k+1})$, this mind change satisfies the strong-monotonicity requirement. Furthermore, M converges to a correct hypothesis for L. Accordingly to the definition, it is easy to see that M is indeed a set-driven IIM. This proves Claim 1, and therefore (1) follows.

In order to prove Assertion (2), we use the following indexed family $\mathcal{L}_{csm} = (L_{\langle k,j\rangle})_{j,k\in\mathbb{N}}$.

For all $k \in \mathbb{N}$ we set $L_{\langle k,0\rangle} = \{a^k b^n \mid n \in \mathbb{N}^+\}$. For all $k \in \mathbb{N}$ and all $j \in \mathbb{N}^+$ we distinguish the following cases:

Case 1. $\neg\Phi_k(k) \le j$

We set $L_{\langle k,j\rangle} = L_{\langle k,0\rangle}$.

Case 2. $\Phi_k(k) \le j$

Then we set $L_{\langle k,j\rangle} = \{a^k b^m \mid 1 \le m \le \Phi_k(k)\} \cup \{a^k b^m \mid m \ge j\}$.

By reducing the halting problem to $\mathcal{L}_{csm} \in \mathit{WMON}$, one may prove that $\mathcal{L}_{csm} \notin \mathit{WMON}$. An IIM M witnessing $\mathcal{L}_{csm} \in$ *s-CSMON* can be easily designed, if one chooses the following hypothesis space $\mathcal{G} = (G_{\langle k,j\rangle})_{j,k\in\mathbb{N}}$. For all $k,\ j \in \mathbb{N}$, we set $L(G_{\langle k,0\rangle}) = \bigcap_{j\in\mathbb{N}} L_{\langle k,j\rangle}$ and $L(G_{\langle k,j+1\rangle}) = L_{\langle k,j\rangle}$. We omit further details.

Furthermore, the family $\mathcal{L}_{csm}$ is witnessing Assertion (3) as well. To see this, recall that *s-CSMON* $\subseteq$ *CLIM* $=$ *ELIM*. Assume any IIM $\hat{M}$ which *ELIM*–identifies $\mathcal{L}$. Since $\mathcal{L}$ contains exclusively infinite languages, it is easy to see that the following set-driven IIM M infers $\mathcal{L}$, too. Let $L \in range(\mathcal{L}_{csm})$, let t be any text for L, and let $x \in \mathbb{N}$. We define:

IIM *M*: "On input t_x do the following: Rearrange the strings contained in t_x^+ in lexicographical order without repetitions. Let $\hat{t}_y = w_0,\ \ldots,\ w_y,\ y \le x$, be the sequence obtained.
If $\hat{M}$ on input $\hat{t}_y$ outputs a hypothesis, then output the same guess and request the next input.
Otherwise, output nothing and request the next input."

The remaining part can be easily shown. One has simply to choose the indexed family used in Lange and Zeugmann [11] to separate *WMON* and *MON*. □

We finish this section with the following corollary.

Corollary 10.

(1) *s-ELIM # ECONSERVATIVE*,

(1) *s-LIM # CONSERVATIVE*.

Proof. By Theorem 9 and Proposition 2 we obtain $s\text{-}ELIM \setminus ECONSERVATIVE \neq \emptyset$ as well as $s\text{-}LIM \setminus CONSERVATIVE \neq \emptyset$. Moreover, Theorem 4 together with Proposition 1 and 2 imply $ECONSERVATIVE \setminus s\text{-}LIM \neq \emptyset$, and hence Assertion (1) and (2) follow. □

4. Learning with Rearrangement-Independent IIMs

In this section we study the impact of rearrangement-independence on the learning power of IIMs. We start with learning in the limit. Angluin [2] characterized the learnability of those indexed families $\mathcal{L}$ that are inferable with respect to the hypothesis space $\mathcal{L}$ in terms of finite, and recursively enumerable tell-tales. Actually, she proved the slightly stronger result that $r\text{-}ELIM = ELIM$. Recently, we showed $ELIM = LIM = CLIM$ (cf. Lange and Zeugmann [14]), and hence we know that rearrangement-independence does not restrict the inference power of IIMs that learn in the limit. However, this general result is also a direct consequence of theorems obtained by Schäfer-Richter [20], and later, but independently by Fulk [5] who proved that any IIM M learning in the limit may be replaced by a rearrangement-independent IIM that infers as least as much than M does. Moreover, Schäfer-Richter's [20] and Fulk's [5] result is much stronger than Angluin's [2], since it is not restricted to the learnability of indexed families. By the next theorem we summarize the known results.

Theorem 11. (Angluin [2], Schäfer-Richter [20], Fulk [5])

$r\text{-}ELIM = ELIM = LIM = CLIM$.

However, neither Schäfer-Richter's [20] nor Fulk's [5] transformation does preserve any of the monotonicity requirements defined above. And indeed, the situation is more subtle than we expected. Furthermore, since the power of all types of monotonic language learning heavily depends on the choice of the hypothesis space, we have to consider separately all the resulting cases. We start with strong-monotonic inference.

Theorem 12.

(1) $r\text{-}ESMON = ESMON$,

(2) $r\text{-}SMON = SMON$.

Proof. First, we prove Assertion (2).

Let $\mathcal{L} \in SMON$. Applying the characterization theorem for $SMON$ (cf. Lange and Zeugmann [10]), we know that there exist a class- preserving hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ as well as a recursively generable family $(T_j)_{j\in\mathbb{N}}$ of finite nonempty sets such that

(i) for all $j \in \mathbb{N}$, $T_j \subseteq L(G_j)$,

(ii) for all $j, k \in \mathbb{N}$, if $T_j \subseteq L(G_k)$, then $L(G_j) \subseteq L(G_k)$.

On the basis of this family $(T_j)_{j\in\mathbb{N}}$ we define an IIM M witnessing $\mathcal{L} \in r\text{-}SMON$. So let $L \in range(\mathcal{L})$, $t \in text(L)$, and $x \in \mathbb{N}$.

IIM *M*: "On input t_x do the following: Search for the least $j \leq x$ for which $T_j \subseteq t_x^+ \subseteq L(G_j)$. If it is found, output j and request the next input.
Otherwise, output nothing and request the next input."

Obviously, M is a rearrangement-independent IIM. It remains to show that M $SMON$–infers $\mathcal{L}$ with respect to the hypothesis space $\mathcal{G}$.

Claim 1. M infers L from text t.

Let $k = \mu z[L(G_z) = L]$. Hence, there is a least x_0 such that $T_k \subseteq t_{x_0}^+$. Therefore, M will output sometimes a hypothesis. For all $j < k$ with $T_j \subseteq L$ we may conclude that $L(G_j) \subseteq L$, since $T_j \subseteq L(G_k)$ implies $L(G_j) \subseteq L(G_k) = L$ (cf. Property (ii)). Moreover, the choice of k yields $L(G_j) \neq L$. Thus, we have $L(G_j) \subset L$. Hence, for every $j < k$ with $T_j \subseteq L$ there exists a y_j such that $t_{y_j}^+ \not\subseteq L(G_j)$. Therefore, $M(t_x) = k$ for all $x > \max\{x_0, y_j \mid\ j < k,\ T_j \subseteq L\}$. This proves Claim 1.

Claim 2. M is strong-monotonic.

Let $M(t_x) = j$ and $M(t_{x+r}) = k$ for some $x \in \mathbb{N}$ and $r \in \mathbb{N}^+$. Due to the definition of M, we have $T_j \subseteq t_x^+ \subseteq L(G_j)$, and $T_k \subseteq t_{x+r}^+ \subseteq L(G_k)$. Therefore, $T_j \subseteq L(G_k)$. Now, Property (ii) implies $L(G_j) \subseteq L(G_k)$. This proves Claim 2.

To sum up, M is witnessing $\mathcal{L} \in r\text{-}SMON$. Thus, Assertion (2) is shown.

Next, we prove Assertion (1). Let $\mathcal{L} \in ESMON$. Because of $ESMON \subseteq SMON$ as well as of Assertion (2), there exist a rearrangement-independent IIM $\hat{M}$ as well as a class-preserving hypothesis space $\mathcal{G}$ such that $\hat{M}$ r-$SMON$–learns $\mathcal{L}$ with respect to the hypothesis space $\mathcal{G}$.

Applying Theorem 4 of Lange and Zeugmann [12], we know that there exists some total recursive function $f\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ satisfying

(iii) for all $j \in \mathbb{N}$, $\lim_{x\to\infty} f(j,x) = k$ exists and satisfies $L(G_j) = L_k$,

(iv) for all $j, x \in \mathbb{N}$, $L_{f(j,x)} \subseteq L_{f(j,x+1)}$.

That means, f is a limiting recursive strong-monotonic compiler from $\mathcal{G}$ into $\mathcal{L}$.

Given the IIM $\hat{M}$, the hypothesis space $\mathcal{G}$ as well as the limiting recursive strong-monotonic compiler f, we define an IIM M witnessing $\mathcal{L} \in r$-$ESMON$. So, let $L \in range(\mathcal{L})$, $t \in text(L)$, and $x \in \mathbb{N}$.

IIM *M*: "On input t_x do the following: Simulate $\hat{M}$ on input t_x. If $\hat{M}$ when successively fed t_x does not output any guess, then output nothing and request the next input. Otherwise, let $j = \hat{M}(t_x)$. If $t_x^+ \subseteq L(G_j)$, then execute (1). Otherwise, output nothing and request the next input.

(1) Find the least $y \in \mathbb{N}$ for which $t_x^+ \subseteq L_{f(j,y)}$. Output $f(j,y)$ and request the next input."

Since the membership problem for $\mathcal{G}$ is uniformly decidable, the test '$t_x^+ \subseteq L(G_j)$' can be effectively performed. Additionally, since $\mathcal{L}$ is an indexed family, the test within Instruction (1) can be effectively accomplished, too. Furthermore, by Property (iii) of f and since $t_x^+ \subseteq L(G_j)$, Instruction (1) has to terminate for every $j \in \mathbb{N}$. Hence, M is indeed an IIM. Due to its definition, M is a rearrangement-independent IIM, since the IIM $\hat{M}$ simulated by M is rearrangement-independent by assumption.

It remains to show that M strong-monotonically infers L from text t. Since $\hat{M}$ infers L from text t and by Property (iii) of f, M converges to a correct hypothesis for L. Finally, we show that M fulfills the strong-monotonicity constraint. Let $f(j,y)$ and $f(k,z)$ denote two hypotheses successively generated by M. Hence, $M(t_x) = f(j,y)$ and $M(t_{x+r}) = f(k,z)$ for some $x \in \mathbb{N}$, $r \in \mathbb{N}^+$. We distinguish the following cases.

Case 1. $j = k$

Due to the definition of M, we may conclude $y \leq z$. Hence, Property (iv) of f guarantees $L_{f(j,y)} \subseteq L_{f(j,z)}$.

Case 2. $j \neq k$

Since f satisfies Properties (iii) and (iv), we obtain $L_{f(j,y)} \subseteq L(G_j)$. Furthermore, M's definition implies $t^+_{x+r} \subseteq L_{f(k,z)}$. Hence, the given IIM $\hat{M}$ has generated the hypothesis j on an initial segment of a text for $L_{f(k,z)} \in \mathcal{L}$. Since $\hat{M}$ behaves strong-monotonically on every text for every language $L \in range(\mathcal{L})$, we may conclude that $L(G_j) \subseteq L_{f(k,z)}$. Together with $L_{f(j,y)} \subseteq L(G_j)$, we get $L_{f(j,y)} \subseteq L_{f(k,z)}$.

Thus, M is rearrangement-independent and strong-monotonic. □

Next we deal with exact and class-preserving monotonic learning.

Theorem 13.

(1) *s-EMON* $\subset$ *r-EMON* $\subset$ *EMON*,

(2) *s-MON* $\subset$ *r-MON* $\subset$ *MON*.

Proof. First of all, we show *r-EMON* $\setminus$ *s-MON* $\neq \emptyset$. By definition, this yields immediately *s-EMON* $\subset$ *r-EMON* as well as *s-MON* $\subset$ *r-MON*.

Lemma 4. *r-EMON* $\setminus$ *s-MON* $\neq \emptyset$.

By Theorem 4 we already know that *r-ESMON* $\setminus$ *s-LIM* $\neq \emptyset$. It is easy to verify that *r-ESMON* $\subseteq$ *r-EMON*. By definition, *s-MON* $\subseteq$ *s-LIM*. Hence, we may conclude *r-EMON* $\setminus$ *s-MON* $\neq \emptyset$. This proves Lemma 4.

It remains to show *EMON* $\setminus$ *r-MON* $\neq \emptyset$. This statement directly implies *r-EMON* $\subset$ *EMON* and *r-MON* $\subset$ *MON*, and hence, the theorem will be proved.

Lemma 5. *EMON* $\setminus$ *r-MON* $\neq \emptyset$.

First of all, we define a corresponding family $\mathcal{L} = (L_k)_{k\in\mathbb{N}}$. For all $k \in \mathbb{N}$ and all $z \in \{0, \dots, 3\}$ we define:

$$L_{4k+z} = \begin{cases} \{a^k b\} \cup A_k, & \text{if } z = 0, \\ \{a^k c\} \cup B_k, & \text{if } z = 1, \\ \{a^k b, a^k c\} \cup A_k, & \text{if } z = 2, \\ \{a^k b, a^k c\} \cup B_k, & \text{if } z = 3. \end{cases}$$

The remaining languages A_k and B_k will be defined via their characteristic functions f_{A_k} and f_{B_k}, respectively. For all $k \in \mathbb{N}$ and all strings $s \in \{a, b, c\}^*$ we set:

$$f_{A_k}(s) = \begin{cases} 1, & \text{if } s = b^k a^m \text{ and } \Phi_k(k) = m, \\ 0, & \text{otherwise.} \end{cases}$$

$$f_{B_k}(s) = \begin{cases} 1, & \text{if } s = c^k a^m \text{ and } \Phi_k(k) = m, \\ 0, & \text{otherwise.} \end{cases}$$

After a little reflection, it is easy to see that $\mathcal{L}$ is indeed an indexed family.

Claim 1. $\mathcal{L} \in EMON$.

We define an IIM which monotonically infers every $L \in range(\mathcal{L})$ from any text t for L with respect to the hypothesis space $\mathcal{L}$ itself. So let us assume any $L \in range(\mathcal{L})$, any text t for L and any $x \in \mathbb{N}$.

IIM *M*: "On input t_x do the following: If $x = 0$ or M has not produced any hypothesis when successively fed t_{x-1}, then execute (1). Otherwise, goto (2).

(1) If $a^k b \in t_x^+$ for some $k \in \mathbb{N}$, then output $4k$ and request the next input.
If $a^k c \in t_x^+$ for some $k \in \mathbb{N}$, then output $4k + 1$ and request the next input.
Otherwise, output nothing and request the next input.

(2) Let $j = M(t_{x-1})$. If $t_x^+ \subseteq L_j$, then repeat the hypothesis j and request the next input.
Otherwise, goto (3).

(3) If $j = 4k$ or $j = 4k+1$, respectively, for some $k \in \mathbb{N}$, then output the hypothesis $j+2$ and request the next input.
If $j = 4k+2$ for some $k \in \mathbb{N}$, output $4k+3$ and request the next input.
Otherwise, output $4k+2$ and request the next input."

It remains to show that $\mathcal{L} \in EMON$ is witnessed by M. Obviously, for every $k \in \mathbb{N}$, M identifies L_{4k} as well as L_{4k+1} from every text for the corresponding language. Thereby, M does not perform any mind change at all. Hence, M is monotonic on every $t \in \mathit{text}(L_{4k}) \cup \mathit{text}(L_{4k+1})$, $k \in \mathbb{N}$. Let us assume any $k \in \mathbb{N}$ such that t is either a text for L_{4k+2} or for L_{4k+3}. In order to show that M satisfies the monotonicity constraint we distinguish the following cases.

Case 1. $\varphi_k(k) \uparrow$

Consequently, we obtain $L_{4k+2} = L_{4k+3}$. Since t is a text for the finite language L_{4k+2}, there is an $x \in \mathbb{N}$ such that $t_x^+ = L_{4k+2}$. Hence, $M(t_{x+r}) = j$ with $L_j = L_{4k+2}$, for all $r \in \mathbb{N}^+$. Furthermore, M has generated at most one different hypothesis before this point. Therefore, M is monotonic. Note that M may converge to different hypotheses on different texts for the same finite language. Consequently, M is not rearrangement-independent.

Case 2. $\varphi_k(k) \downarrow$

Since L_{4k+2} as well as L_{4k+3} define finite languages, it is easy to see that M converges to a correct hypothesis. We distinguish the following subcases.

Subcase 2.1. t is a text for L_{4k+2}.

If M first generates the hypothesis $4k$, then it needs only one mind change to infer L_{4k+2}. Consequently, M is monotonic. Otherwise, $4k+1$ is M's first hypothesis. Now, it is easy to verify that M produces the sequence of hypotheses $4k+1$, $4k+3$ and $4k+2$. Due to the definition of the family $\mathcal{L}$, $L_{4k+1} \cap L_{4k+2} \subset L_{4k+3}$ directly implies $L_{4k+1} \cap L_{4k+2} \subseteq L_{4k+3} \cap L_{4k+2} \subseteq L_{4k+2}$. Hence, M is again monotonic.

Subcase 2.2. t is a text for L_{4k+3}.

Then, a quite similar argumentation yields that M fulfills the monotonicity constraint. If M outputs the hypothesis $4k+1$ as its first guess, then again, one mind change suffices to identify L_{4k+3}. Otherwise, M produces the sequences of hypotheses $4k$, $4k+2$ and $4k+3$. Due to the definition of the family $\mathcal{L}$, $L_{4k} \cap L_{4k+3} \subset L_{4k+2}$. As before, this directly implies that M is monotonic.

Therefore, M witnesses $\mathcal{L} \in EMON$, and Claim 1 is proved.

Claim 2. $\mathcal{L} \notin r\text{-}MON$.

Suppose the converse, i.e., there are a class-preserving hypothesis space $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ and an IIM M that witnesses $\mathcal{L} \in r\text{-}MON$ with respect to $\mathcal{G}$.

Claim 3. Given $\mathcal{G}$ and any program for M witnessing $\mathcal{L} \in r\text{-}MON$, one can effectively construct an algorithm deciding whether or not $\varphi_k(k)$ converges.

Next, we define the desired algorithm.

Algorithm $\mathcal{C}$: "On input k execute (C1) until (α1) or (α2) is fulfilled, respectively. Afterwards, execute (C2).

(C1) For all $x = 0, 1, \ldots$, execute in parallel (α1) and (α2) until one of them is successful.

(α1) Test whether $\Phi_k(k) \le x$.

(α2) Simulate M when fed the initial segments t_x and $\hat{t}_x$ of the uniquely defined texts for $L = \{a^k b\}$ and $\hat{L} = \{a^k c\}$, respectively. If M outputs on both initial segments a hypothesis, say m and $\hat{m}$, respectively, then test whether $a^k b \in L(G_m)$, $a^k c \notin L(G_m)$, $a^k c \in L(G_{\hat{m}})$, and $a^k b \notin L(G_{\hat{m}})$.

(C2) If (α1) happens first, then output '$\varphi_k(k)$ converges' and stop.

Otherwise, in parallel execute (β1) or (β2) for $y = 1, 2, \ldots$, until one of them is successful.

(β1) Test whether $\Phi_k(k) \le x + y$.

(β2) Test whether M, when fed $t_{x+y} = \underbrace{a^k b, \ldots, a^k b}_{(x+1)-times}, \underbrace{a^k c, \ldots, a^k c}_{y-times}$, generates a consistent hypothesis n, i.e., $M(t_{x+y}) = n$ and $t^+_{x+y} \subseteq L(G_n)$.

If (β1) happens first, then output '$\varphi_k(k)$ converges' and stop.

Otherwise, output '$\varphi_k(k)$ diverges' and stop."

Due to the definition of a complexity measure, (α1) and (β1) can be effectively accomplished. Furthermore, since M is an IIM, and since membership is uniformly decidable for $\mathcal{L}(\mathcal{G})$, (α2) and (β2) can be effectively accomplished, too. Hence, $\mathcal{C}$ is indeed an algorithm.

First, we show that $\mathcal{C}$ terminates for all $k \in \mathbb{N}$. Let us assume that the execution of (C1) does not terminate for some $k \in \mathbb{N}$. Obviously, then $\varphi_k(k)$ diverges. Consequently, $L_{4k} = L$ and $L_{4k+1} = \hat{L}$. Now, since (α2) will never terminate successfully, M fails to infer at least one of the languages $\{a^k b\}$, $\{a^k c\}$ from its uniquely defined text, a contradiction. Applying the same argument one can show that the execution of (C2) has to terminate, too. Hence, $\mathcal{C}$ terminates on every input $k \in \mathbb{N}$.

It remains to show that $\mathcal{C}$ works correctly. Obviously, if $\mathcal{C}$ stops its computation with '$\varphi_k(k)$ converges,' then $\varphi_k(k)$ is indeed defined. Suppose that $\mathcal{C}$ has finished its computation with '$\varphi_k(k)$ diverges.' Furthermore, assume that $\varphi_k(k)$ is defined. Due to our definition, there exists an $x \in \mathbb{N}$ such that $M(t_x) = m$ and $M(\hat{t}_x) = \hat{m}$ with $t_x^+ \subseteq L(G_m)$ as well as $\hat{t}_x^+ \subseteq L(G_{\hat{m}})$. Since $\mathcal{G}$ is a class-preserving hypothesis space, $L(G_m) = L_{4k}$ and $L(G_{\hat{m}}) = L_{4k+1}$, respectively, follows immediately (cf. $\mathcal{L}$'s definition). Additionally, there exists a $y \in \mathbb{N}^+$ such that $M(t_{x+y}) = n$ with $t_{x+y}^+ \subseteq L(G_n)$. Since M is rearrangement-independent, we may conclude that $M(\hat{t}_{x+y}) = n$, where $\hat{t}_{x+y} = \underbrace{a^k c, \ldots, a^k c}_{(x+1)-times}, \underbrace{a^k b, \ldots, a^k b}_{y-times}$, since $t_{x+y}^+ = \hat{t}_{x+y}^+$. Because $\{a^k b, a^k c\} \subseteq L(G_n)$ as well as $L(G_n) \in range(\mathcal{L})$, it suffices to distinguish the following two cases.

Case 1. $L(G_n) = L_{4k+2}$

Clearly, $\hat{t}_{x+y}$ is an initial segment of a text for L_{4k+3}, too. On this text, M has already generated the hypotheses $\hat{m}$ and n in some subsequent steps. Since $\varphi_k(k)$ is defined, by assumption we obtain $L_{4k+1} \cap L_{4k+3} \not\subseteq L_{4k+2} \cap L_{4k+3}$. Therefore, $L(G_{\hat{m}}) = L_{4k+1}$ and $L(G_n) = L_{4k+2}$ directly imply that M violates the monotonicity requirement, a contradiction.

Case 2. $L(G_n) = L_{4k+3}$

Using similar arguments, it is easy to see that M violates the monotonicity requirement when inferring L_{4k+2} from any of its texts having the initial segment t_{x+y}.

This proves the correctness of Algorithm $\mathcal{C}$. Thus, Claim 3 is shown.

On the other hand, the halting problem is undecidable. Therefore, Claim 2 follows, and Lemma 5 is proved. □

Finally, we consider rearrangement-independence in the context of exact and class-preserving conservative learning. Since conservative learning is exactly as powerful as weak-monotonic one, by the latter Theorem it might be expected that rearrangement-independence is a severe restriction under the weak-monotonic constraint, too. On the other hand, looking at Theorem 6 we see that conservative learning has its peculiarities. And indeed, exact and class-preserving learning can always be performed by rearrangement-independent IIMs. In order to prove this, we first characterize *ECONSERVATIVE* in terms of finite tell-tales. We present this theorem separately, since it is interesting in its own right.

Theorem 14. *Let $\mathcal{L}$ be an indexed family. Then, $\mathcal{L} \in ECONSERVATIVE$ if and only if there exists a recursively generable family $(T_j^y)_{j,y\in\mathbb{N}}$ of finite sets such that*

(1) *for all $L \in range(\mathcal{L})$ there exists a j with $L_j = L$ and $T_j^y \neq \emptyset$ for almost all $y \in \mathbb{N}$,*

(2) *for all $j, y \in \mathbb{N}$, $T_j^y \neq \emptyset$ implies $T_j^y \subseteq L_j$ and $T_j^y = T_j^{y+1}$,*

(3) *for all $j,\ y,\ z \in \mathbb{N}$, $\emptyset \neq T_j^y \subseteq L_z$ implies $L_z \not\subset L_j$.*

Proof. Necessity. Let $\mathcal{L} \in ECONSERVATIVE$. Then there is an IIM M that conservatively learns $\mathcal{L}$ with respect to $\mathcal{L}$. The desired tell-tale family $(T_j^y)_{j,y\in\mathbb{N}}$ is defined as follows. Let $j, y \in \mathbb{N}$; then we set

$$T_j^y = \begin{cases} range(t_z^j) \text{ with } z = min\{x \mid\ x \leq y,\ M(t_x^j) = j\}, & \text{if there is an } x \leq y \\ & \text{such that } M(t_x^j) = j \\ \emptyset, & \text{otherwise,} \end{cases}$$

where t^j denotes the canonical text of L_j. Obviously, the sets T_j^y are uniformly recursively generable and finite. It remains to show that the Properties (1) through (3) are fulfilled.

By construction, (2) is trivially satisfied. In order to prove (1), let $L \in range(\mathcal{L})$ and let t^L be the canonical text of L. Since M has to infer L from its canonical text, too, there exists a j such that $j = M(t_x^L)$ for almost all $x \in \mathbb{N}$ and $L = L_j$. Let $y = \mu x[M(t_x^L) = j]$. Then $T_j^y \neq \emptyset$ and $T_j^y = T_j^{y+r}$ for all $r \in \mathbb{N}$. This proves Property (1). Finally, we have to show (3). Suppose, there are $j, y, z \in \mathbb{N}$ such that $\emptyset \neq T_j^y \subseteq L_z$ and $L_z \subset L_j$. By the definition of the tell-tale sets, there exists an $x \leq y$ such that M on input t_x^j outputs j. Furthermore, $T_j^y \subseteq L_z$ and therefore, t_x^j is an initial segment of a text for L_z, too. Since M has to infer L_z from every text, it has to perform at least one mind change on every text $t \in text(L_z)$ beginning with t_x^j that cannot be caused by an inconsistency. This contradiction proves (3).

Sufficiency. The desired IIM is defined as follows. Let $L \in range(\mathcal{L})$, $t \in text(L)$, and $x \in \mathbb{N}$.

IIM *M*: "On input t_x do the following: If $x = 0$ or $x > 0$ and M on input t_{x-1} does not produce any hypothesis, then goto (B).
Otherwise, goto (A).

(A) Let j be M's last hypothesis on input t_{x-1}. Test whether or not $t_x^+ \subseteq L_j$. In case it is, output j and request the next input.
Otherwise, goto (B).

(B) Generate T_j^y for all $j, y = 1, ..., x$ and test whether or not $T_j^y \neq \emptyset$. For all nonempty T_j^y check whether or not $T_j^y \subseteq t_x^+ \subseteq L_j$. In case there is one j fulfilling the test, output the minimal one, and request the next input.
Otherwise, output nothing and request the next input."

Using the same arguments as in the proof of Theorem 1 in Lange and Zeugmann [10], it is easy to see that M *ECONSERVATIVE*–learns $\mathcal{L}$. We omit the details. □

Now we are ready to prove the announced theorem stating that rearrangement-independence does not restrict exact and class-preserving conservative learning.

Theorem 15.

(1) *r-ECONSERVATIVE* = *ECONSERVATIVE*,

(2) *r-CONSERVATIVE* = *CONSERVATIVE*.

Proof. First we prove Assertion (1). Let $\mathcal{L} \in \mathit{ECONSERVATIVE}$. By Theorem 14 there exists a recursively generable family $(T_j^y)_{j,y\in\mathbb{N}}$ fulfilling Properties (1) through (3). Using this family, we define a new recursively generable family $(\hat{T}_j^y)_{j,y\in\mathbb{N}}$ that satisfies (1), (2), and (3), too. However, the new family allows the design of a rearrangement-independent IIM, while the IIM described in the proof of Theorem 14 is not rearrangement-independent.

We set $\hat{T}_j^y = \emptyset$, if $j \geq y$. Now, let $j < y$; we define

$$\hat{T}_j^y = \begin{cases} \hat{T}_j^{y-1}, & \text{if } \hat{T}_j^{y-1} \neq \emptyset, \\ \emptyset, & \text{if } \hat{T}_j^{y-1} = \emptyset,\ T_j^y = \emptyset, \\ \bigcup_{k\leq y-1} T_k^y \cap L_j, & \text{otherwise.} \end{cases}$$

Since $(T_j^y)_{j,y\in\mathbb{N}}$ is a recursively generable family and because of the uniform decidability of the membership problem for $\mathcal{L}$, the family $(\hat{T}_j^y)_{j,y\in\mathbb{N}}$ is recursively generable, too. It is easy to see that $(\hat{T}_j^y)_{j,y\in\mathbb{N}}$ fulfills Properties (1) through (3) of Theorem 14. We proceed with a technical claim that will be very useful in proving the rearrangement-independence of the IIM defined below.

Claim 1. Let $j, k, m, n \in \mathbb{N}$ such that $m = \mu y[\hat{T}_j^y \neq \emptyset]$ and $n = \mu y[\hat{T}_k^y \neq \emptyset]$. Then, $\hat{T}_k^n \cap L_j \subseteq \hat{T}_j^m$ provided $n \leq m$.

In accordance with the definition of the family $(\hat{T}_j^y)_{j,y\in\mathbb{N}}$ we know that $m > j$ as well as $n > k$. Furthermore, $\hat{T}_j^m = \bigcup_{z\leq m-1} T_z^m \cap L_j$ and $\hat{T}_k^n = \bigcup_{z\leq n-1} T_z^n \cap L_k$. Hence, we obtain:

$$\hat{T}_k^n \cap L_j = \bigcup_{z\leq n-1} T_z^n \cap L_k \cap L_j \subseteq \bigcup_{z\leq n-1} T_z^n \cap L_j \subseteq \bigcup_{z\leq n-1} T_z^m \cap L_j \subseteq \bigcup_{z\leq m-1} T_z^m \cap L_j = \hat{T}_j^m.$$

This proves Claim 1.

Now we define the desired rearrangement-independent IIM as follows. Let $L \in \mathit{range}(\mathcal{L})$, $t \in \mathit{text}(L)$, and $x \in \mathbb{N}$.

IIM *M*: "On input t_x do the following: Test for all $k \leq x$ whether or not $\hat{T}_k^x \neq \emptyset$. For all nonempty $\hat{T}_k^x$ check whether or not $\hat{T}_k^x \subseteq t_x^+ \subseteq L_k$.

In case there is no k fulfilling the test, output nothing and request the next input.

Otherwise, compute $y_k = \mu y[\hat{T}_k^y \neq \emptyset]$ for all k fulfilling the test. Output the minimal k for which y_k is minimal, and request the next input."

It remains to show that M witnesses $\mathcal{L} \in$ *r-ECONSERVATIVE*. Obviously, M is rearrangement-independent.

Claim 2. M is conservative.

Let j and k, $j \neq k$, be two hypotheses produced by M on input t_x and t_{x+r}, respectively. We have to show that $t_{x+r}^+ \not\subseteq L_j$. In accordance with M's definition we directly obtain $\hat{T}_j^x \neq \emptyset \neq \hat{T}_k^{x+r}$. We consider the following cases.

Case 1. $\hat{T}_k^x = \emptyset$.

Then we have $t_{x+r}^+ \not\subseteq L_j$. This can be seen as follows. M on input t_{x+r} has to compute y_j and y_k. Since $\hat{T}_k^x = \emptyset$, we know that $y_j < y_k$. Consequently, if $t_{x+r}^+ \subseteq L_j$, then M outputs j, a contradiction.

Case 2. $\hat{T}_k^x \neq \emptyset$.

Let $m = \mu y[\hat{T}_j^y \neq \emptyset]$ and $n = \mu y[\hat{T}_k^y \neq \emptyset]$. We distinguish the following subcases.

Subcase 2.1. $m < n$

Applying the same arguments as in Case 1 directly yields $t_{x+r}^+ \not\subseteq L_j$.

Subcase 2.2. $m = n$

Suppose $j < k$. Again, by the same arguments as in Case 1 one directly obtains $t_{x+r}^+ \not\subseteq L_j$. We proceed with $k < j$. By Claim 1 we get $\hat{T}_k^n \cap L_j \subseteq \hat{T}_j^m$. Suppose, $t_{x+r}^+ \subseteq L_j$. Since $k = M(t_{x+r})$, we immediately obtain that $\hat{T}_k^n \subseteq t_{x+r}^+ \subseteq L_k$. Consequently, $\hat{T}_k^n \cap L_j = \hat{T}_k^n$, and hence $\hat{T}_k^n \subseteq \hat{T}_j^m \subseteq t_x^+$, since $j = M(t_x)$. But this implies $M(t_x) = k$, since $j > k$, a contradiction.

Subcase 2.3.: $m > n$

Again, by Claim 1 we know that $\hat{T}_k^n \cap L_j \subseteq \hat{T}_j^m$. Moreover, by assumption $j = M(t_x)$, and therefore $\emptyset \neq \hat{T}_j^m \subseteq t_x^+$. Because of $m > n$, we furthermore conclude that $\hat{T}_k^n \not\subseteq t_x^+$, since otherwise $M(t_x) = k$. On the other hand, $\hat{T}_k^n \subseteq t_{x+r}^+$, since $\hat{T}_k^n \not\subseteq t_{x+r}^+$ directly implies $k \neq M(t_{x+r})$. Finally, if $t_{x+r}^+ \subseteq L_j$, then $\hat{T}_k^n \cap L_j = \hat{T}_k^n \subseteq \hat{T}_j^m$. But this would imply $\hat{T}_k^n \subseteq t_x^+$, again a contradiction.

Hence, M is conservative, and Claim 2 is proved.

Claim 3. M infers $\mathcal{L}$.

Let $L \in range(\mathcal{L})$ and let $t \in text(L)$. Moreover, let $K = \{k \mid\ L_k = L,\ \hat{T}_k^y \neq \emptyset$ for almost all $y \in \mathbb{N}\}$. By Property (1) of the family $(\hat{T}_j^y)_{j,y\in\mathbb{N}}$ we know that $K \neq \emptyset$. Let $k = \min K$, and let $y_k = \mu y[\hat{T}_k^y \neq \emptyset]$. Since t is a text for L, there exists an $x \geq y_k$ such that $\hat{T}_k^{y_k} \subseteq t_x^+ \subseteq L_k = L$. Hence, on every input t_{x+r} the IIM M has to output a hypothesis. We consider the set C of possible hypotheses that might be output by M after having read $t_{x+r},\ r \geq 0$. Let $y_j = \mu y[\hat{T}_j^y \neq \emptyset]$, then C may be written as $C = \{j \mid\ j \in \mathbb{N},\ y_j \leq y_k,\ \hat{T}_j^{y_j} \subseteq range(t)\}$. Due to the definition of the family $(\hat{T}_j^y)_{j,y\in\mathbb{N}}$ the condition $\hat{T}_j^y \neq \emptyset$ implies $y < j$. Therefore, $j > y_k$ directly yields $y_j > y_k$. Hence, we may rewrite C as $C = \{j \mid\ j \leq y_k,\ y_j \leq y_k,\ \hat{T}_j^{y_j} \subseteq range(t)\}$. Consequently, C is finite. Finally, applying Property (3) of Theorem 14 we may conclude $L_k \not\subset L_j$ for all $j \in C$ with $j \neq k$. Hence, for all $j \in C$ with $L_j \neq L_k$ there exists an x_j such that $t_{x_j}^+ \not\subseteq L_j$. Since C is finite, it successively shrinks to $\hat{C} = \{k\} \cup \{\ell \mid\ \ell \leq y_k,\ L_\ell = L_k\}$, and hence M sometimes outputs an element $\hat{k}$ from $\hat{C}$. Since M is conservative (see Claim 2), M has to repeat $\hat{k}$ in every subsequent step, and thus it converges.

This proves Assertion (1) of the theorem. Assertion (2) can be proved analogously as Lemma 1 in the proof of Theorem 6. □

5. Summary

In the present paper we have studied the question whether or not the *order* of information presentation does really influence the capabilities of learning algorithms that infer *indexed families* from *positive data*. On the one hand, it has been known that set-driveness constitutes a severe restriction for learning classes of recursively enumerable languages (cf. Schäfer-Richter [20], and Fulk [5]). On the other hand, the knowledge concerning the learnability of indexed families has been much more restricted. Our results provide a thorough analysis concerning the learning capabilities of set-driven and rearrangement-independent IIMs in the setting of learning indexed families. In particular, the theorems obtained relate the learning power of IIMs that satisfy the subset principle to different extends and of IIMs that simultaneously learn order independently to one another. Moreover, our results provide strong evidence that the choice of the relevant hypothesis space is of particular importance if one is interested in superior learning algorithms. In this regard, the Theorems 6, 8 as well as Corollary 7 may be considered as a partial answer to the problem of what a natural learning algorithm should look like. Putting them all together we obtain the following. An indexed family $\mathcal{L}$ can be class-comprisingly inferred in the limit by a set-driven IIM if and only if there are a hypothesis space $\mathcal{G}$ with $range(\mathcal{L}) \subseteq range(\mathcal{L}(\mathcal{G}))$ and a conservative and set-driven IIM that learns all languages from $\mathcal{L}(\mathcal{G})$. Moreover, since s-$CLIM$ equals $CCONSERVATIVE$, the latter result may simplify the design of superior learning algorithms.

In a preliminary version of this paper we left it open to what extend set-driveness and rearrangement-independence, respectively, influences the capabilities of class-comprising strong-monotonic and monotonic learning. Inspired by our results, Stephan [21] completely solved these questions. In particular, he proved that s-$CSMON = CSMON$ and s-$CMON \subset r$-$CMON = CMON$.

We continue with the following figure that summarizes our results and the results obtained by Stephan [21].

	exact learning	class-preserving learning	class-comprising learning
FIN	*set drivenness* +	*set drivenness* +	*set drivenness* +
SMON	*rearrangement independence* +	*rearrangement independence* +	*set drivenness* +
MON	*rearrangement independence* −	*rearrangement independence* −	*rearrangement independence* +
WMON	*rearrangement independence* +	*rearrangement independence* +	*set drivenness* +
LIM	*rearrangement independence* +	*rearrangement independence* +	*rearrangement independence* +

For every mode of learning ID mentioned "*rearrangement-independence* +" indicates r-$ID =$

ID as well as $s\text{-}ID \subset ID$. "*Rearrangement-independence* $-$" implies $s\text{-}ID \subset r\text{-}ID \subset ID$ whereas "*set-drivenness* $+$" should be interpreted as $s\text{-}ID = ID$ and, therefore, $r\text{-}ID = ID$, too.

Let us continue with an intuitive explanation of the results displayed in the figure above. As the proof of Theorem 1 shows finite learning can always be achieved by set-driven IIMs that are allowed to perform *unbounded* search in the relevant hypothesis space. The unbounded search for the first consistent hypothesis is combined with a test for equality enabling the IIM to decide whether or not it has reached its learning goal. The effectiveness of the equality test is guaranteed by the properties of the relevant family of uniformly recursively generable finite sets that reflect the topological structure of the target indexed family.

Looking at strong-monotonic learning the situation considerably changes. As long as exact and class-preserving learning are concerned unbounded search is no longer an option. Instead, the actual input length is used to *bound* the actual set of admissible hypotheses. However, the relevant family of uniformly recursively generable finite sets is still the main ingredient to solve the subset problem (cf. Theorem 12). As a result, exact and class-preserving strong-monotonic learning can be always performed by rearrangement-independent IIMs but in general not by set-driven learning devices. As Stephan [21] showed, class-comprising strong-monotonic inference can be generally realized by set-driven IIMs. Again, these learning algorithms perform *unbounded* search within an appropriate hypothesis space.

As the above figure shows monotonic learning has its peculiarities. Intuitively, the main difficulty any monotonic IIM has to handle can be described as follows. The demand to its output is closely related to the *whole* target language. However, at each learning step the IIM has only access to the finite subset of the target language that it provided as input. Furthermore, that the intersection of the hypothesized language and the target language may yield a language outside the class-preserving hypothesis space has to be taken into account. Now it becomes evident that every monotonic IIM is in serious trouble whenever it has no access to its previously performed learning steps. Clearly, the requirement to learn rearrangement-independently or even set-drivenly complicates any such access. And indeed, as the proof of Theorem 13 shows the resulting information loss may lead to the unsolvability of the monotonic learning task.

Concerning conservative learning the situation again slightly differs. The main reason for that phenomenon can be intuitively explained by the replacement of *global* demands on the created hypotheses by *local* ones. If the actual hypothesis is consistent, then the state of the art already reached during the learning process is expressed with sufficient accuracy. However, this only true as long as the subset problem can be effectively handled. Hence, Theorem 8 impressively shows the equivalence of set-drivenness and the recursive solvability of the subset problem for the resolution of a learning problem. However, at this point, the choice of the relevant hypothesis space again becomes important, since the descriptions and enumeration chosen may heavily influence the algorithmic solvability of the subset problem.

Next, we point to further problems that are closely related to the impact of order independence to learning. First, it would be highly desirable to elaborate characteristic conditions under what circumstances set-drivenness does not restrict the learning power. We expect that such characterizations might allow much more insight into the problem how to handle simultaneously both, finite and infinite languages in the learning process. Next, as we have seen, an algorithmically solvable learning problem might become infeasible, if one tries to solve it with set-driven IIMs. On the other hand, when dealing with particular learning problems it might often be possible to design a set-driven learning algorithm solving it. But what about the complexity of learning in such circumstances? More precisely, we are interested in knowing whether the "high-level" theorem separating set-driven learning from unrestricted one, has an analogue in terms of complexity theory. For example, it is well conceivable that an indexed family $\mathcal{L}$ may be learned in polynomial time but no set-driven algorithm can efficiently infer $\mathcal{L}$ provided $\mathcal{P} \neq \mathcal{NP}$. Recently, a similar approach has been successfully undertaken concerning consistent and inconsistent learning algorithms (cf. Wiehagen and Zeugmann [25]).

Finally, many researchers considered the learnability of languages from both *positive* and *negative data* (cf., e.g., Gold [6], Lange and Zeugmann [15], Mukouchi [18], Wiehagen and Zeugmann [25]). In this setting, a learner is successively fed all strings over the underlying alphabet which are classified with respect to their containment in the target language. Hence, it is only natural to ask whether or not order is of the same importance in this model. Concerning learning in the limit the negative answer has been provided by Blum and Blum [3] for the general case of learning recursively enumerable languages. However, the equality of set-driven learning in the limit and unrestricted identification in the limit for indexed families goes back to Gold [6]. We refer the reader to Zeugmann and Lange [27] for a more detailed discussion. Moreover, these results extend to all the models of monotonic language learning from positive and negative data. In particular, by characterizing all types of monotonic language learning from positive and negative data in terms of uniformly recursively generable families of positive and negative finite sets Lange and Zeugmann [15] showed the following. There is exactly one learning algorithm that can perform every learning task under the relevant monotonicity constraint. As a matter of fact, this algorithm is set-driven, too.

Acknowledgement

Most of this work has been done while the first author was visiting ISIS FUJITSU Labs. at Numazu, Japan. He is greatfully indebted to Y. Takada and Y. Sakakibara for providing inspiring working conditions.

Furthermore, the first author has been partially supported by the German Ministry for Research and Technology (BMFT) under grant no. 01 IW 101.

Finally, we heartily thank the anonymous referees for their careful reading and for many valuable comments on the preparation of this paper.

6. References

[1] D. Angluin, Finding patterns common to a set of strings, *Journal of Computer and System Sciences* **21** (1980), 46 – 62.

[2] D. Angluin, Inductive inference of formal languages from positive data, *Information and Control* **45** (1980), 117 – 135.

[3] L. Blum and M. Blum, Toward a mathematical theory of inductive inference, *Information and Control* **28** (1975), 122 – 155.

[4] R. Berwick, *The Acquisition of Syntactic Knowledge*, MIT Press, Cambridge, Mass., 1985.

[5] M. Fulk, Prudence and other restrictions in formal language learning, *Information and Computation* **85** (1990), 1 – 11.

[6] E.M. Gold, Language identification in the limit, *Information and Control* **10** (1967), 447 – 474.

[7] J.E. Hopcroft and J.D. Ullman, *Formal Languages and their Relation to Automata*, Addison-Wesley, Reading, Mass., 1969.

[8] K.P. Jantke, Monotonic and non-monotonic inductive inference, *New Generation Computing* **8** (1991), 349 – 360.

[9] S. Lange and R. Wiehagen, Polynomial-time inference of arbitrary pattern languages, *New Generation Computing* **8** (1991), 361 – 370.

[10] S. Lange and T. Zeugmann, Types of monotonic language learning and their characterization, *Proc. 5th Annual ACM Workshop on Computational Learning Theory*, ACM Press, New York, 1992, pp. 377 – 390.

[11] S. Lange and T. Zeugmann, Monotonic versus non-monotonic language learning, *Proc. 2nd International Workshop on Nonmonotonic and Inductive Logic* (G. Brewka, K.P. Jantke and P.H. Schmitt, Eds.), Springer-Verlag, Berlin, 1993, Lecture Notes in Artificial Intelligence 659, pp. 254 – 269.

[12] S. Lange and T. Zeugmann, Language learning in dependence on the space of hypotheses, *Proc. 6th Annual ACM Conference on Computational Learning Theory*, ACM Press, New York, 1993, pp. 127 – 136.

[13] S. Lange and T. Zeugmann, The learnability of recursive languages in dependence on the hypothesis space, HTWK Leipzig, FB Mathematik, Informatik und Naturwissenschaften, GOSLER–Report 20/93, 1993.

[14] S. Lange and T. Zeugmann, Learning recursive languages with bounded mind changes, *International Journal of Foundations of Computer Science* **4** (1993), 157 – 178.

[15] S. Lange and T. Zeugmann, Characterization of language learning from informant under various monotonicity constraints, *Journal of Experimental and Theoretical Artificial Intelligence* **6** (1994), 71 – 94.

[16] S. Lange, T. Zeugmann and S. Kapur, Monotonic and dual-monotonic language learning, *Theoretical Computer Science* **155** (1996), 365 – 410.

[17] M. Machtey and P. Young, *An Introduction to the General Theory of Algorithms*, North-Holland, New York, 1978.

[18] Y. Mukouchi, Inductive inference with bounded mind changes, *Proc. 3rd Workshop on Algorithmic Learning Theory* (S. Doshita, K. Furukawa, K.P. Jantke and T. Nishida, Eds.), Springer-Verlag, Berlin, 1993, Lecture Notes in Artificial Intelligence 743, pp. 125 – 134.

[19] 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.

[20] G. Schäfer-Richter, Über Eingabeabhängigkeit und Komplexität von Inferenzstrategien, Rheinisch Westfälische Technische Hochschule Aachen, Dissertation, 1984.

[21] F. Stephan, Personal communication

[22] K. Wexler, The subset principle is an intensional principle, In *Knowledge and Language: Issues in Representation and Acquisition* (E. Reuland and W. Abraham, Eds.), Vol. 1, From Orwell's Problem to Plato's Problem, Chapter 9, Kluwer Academic Publishers, 1993.

[23] K. Wexler and P. Culicover, *Formal Principles of Language Acquisition*, MIT Press, Cambridge, Mass., 1980.

[24] R. Wiehagen, A thesis in inductive inference, *Proc. First International Workshop on Nonmonotonic and Inductive Logic* (J. Dix, K.P. Jantke and P.H. Schmitt, Eds.), Springer-Verlag, Berlin, 1991, Lecture Notes in Artificial Intelligence 543, pp. 184 – 207.

[25] R. Wiehagen and T. Zeugmann, Ignoring data may be the only way to learn efficiently, *Journal of Experimental and Theoretical Artificial Intelligence* **6** (1994), 131 – 144.

[26] T. Zeugmann, Lange and Wiehagen's pattern language learning algorithm: An average-case analysis with respect to its total learning time, RIFIS Technical Report RIFIS-TR-CS-111, RIFIS, Kyushu University 33, 1995.

[27] T. Zeugmann and S. Lange, A guided tour across the boundaries of learning recursive languages, *Algorithmic Learning for Knowledge-Based Systems* (K.P. Jantke and S. Lange, Eds.), Springer-Verlag, Berlin, 1995, Lecture Notes in Artificial Intelligence 961, pp. 190 – 258.