On the Impact of Order Independence to the Learnability of Recursive Languages

Steffen Lange*
HTWK Leipzig
FB Mathematik und Informatik
PF 66
04275 Leipzig
steffen@informatik.th-leipzig.de

Thomas Zeugmann
TH Darmstadt
Institut für Theoretische Informatik
Alexanderstr. 10
64283 Darmstadt
zeugmann@iti.informatik.th-darmstadt.de

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

Abstract

The present paper deals with the learnability of indexed families of uniformly recursive languages from positive data under various postulates of naturalness. 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. 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 twofold. First, rearrangement-independent learning does not constitute a restriction except the case of monotonic learning. Second, we prove that for all but one of the considered learning models set-drivenness is a severe restriction. However, set-driven *conservative* learning is exactly as powerful as unrestricted *conservative* learning provided the *hypothesis space* is appropriately chosen. These results considerably extend previous work done in the field (cf. e.g. Schäfer (1984) and Fulk (1990)).

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

*Most of this work has been done while the author was visiting ISIS FUJITSU Labs. at Numazu, Japan. It has been partially supported by the German Ministry for Research and Technology (BMFT) under grant no. 01 IW 101.

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 (1967) 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 (1986) and the references therein). Most of the work done in the field has been aimed at two goals: the characterization of those collections of languages that can be learned, and to study the impact of several postulates on the behavior of learners to their learning power.

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 next to 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 hypothesis 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 $\mathcal{G}$.

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 does only depend on the *range* of its input. Surprisingly enough, Schäfer (1984) and Fulk (1990) proved that set-driven IIMs are less powerful than unrestricted ones. Intuitively, the weakness of set-driven IIMs is caused by the difficulties to handle both, finite and infinite languages. A natural weakening of set-drivenness is rearrangement-independence. An IIM is called *rearrangement-independent* if its output does only depend 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 (1984), Fulk (1990)). However, the weakness of set-driven IIMs has been proved in a setting allowing self-referential arguments. This might lead to the impression that this result is far beyond any practical relevance, since self-referential arguments are exclusively applicable in settings where the *membership problem* for languages is undecidable in general.

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 of non-empty and uniformly recursive languages. A famous example for an indexed family is the collection of all pattern languages (cf. Angluin (1980a)). Although this indexed families contains finite and infinite languages, Lange and Wiehagen (1991) succeeded in designing a set-driven IIM learning it. 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, one has to deal 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's (1984) nor Fulk's (1990) 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 (1993b)). Hence, we have to distinguish between exact learning, class preserving inference, and class comprising learning. 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 to 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 (1985), Wexler (1992)). Informally, the subset principle requires the learner to hypothesize the "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 has been introduced by Jantke (1991) and monotonic learning goes back to Wiehagen (1991). Subsequently, we have studied their learning capabilities in the setting of learning indexed families (cf. Lange and Zeugmann (1993a)). Again, the power of all the monotonic learning models heavily depends on the choice of the hypothesis space (cf. Lange and Zeugmann (1993b).

In the sequel 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. Moreover, we show that set-drivenness cannot be achieved in general. However, class comprising weak-monotonic learning is exactly as powerful than class comprising set-driven weak-monotonic inference. We regard this result as a particular answer to the question 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

$\varphi_0, \varphi_1, \varphi_2, \ldots$ denote any fixed programming system of all (and only all) partial recursive functions over $\mathbb{N}$, and let $\Phi_0, \Phi_1, \Phi_2, \ldots$ be any associated complexity measure (cf. Machtey and Young, 1978). 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 $\langle ., . \rangle : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ we denote Cantor's pairing function. Moreover, we use $\langle ., ., . \rangle : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ to denote the following encoding: $\langle x, y, z \rangle =_{df} \langle x, \langle y, z \rangle \rangle$ for all $x, y, z \in \mathbb{N}$.

In the sequel we assume familiarity with formal language theory (cf. Hopcroft and Ullman (1969)). By Σ we denote any fixed finite alphabet of symbols. Let Σ^* be the free monoid over Σ. Any subset $L \subseteq \Sigma^*$ is called a language. By $co-L$ we denote the complement of L. Let L be a language and $t = s_0, s_1, s_2, \ldots$ an infinite sequence of strings from Σ^* such that $range(t) = \{s_k \,|\, k \in \mathbb{N}\} = L$. 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 $t_x^+ =_{df} \{s_k \,|\, k \leq x\}$.

Next, we introduce the notion of *canonical text* that turned out to be very helpful in proving several theorems. Let L be any non-empty recursive language, and let $s_0, s_1, s_2, \ldots$ be the lexicographically ordered text of Σ^*. The canonical text of L is obtained as follows. Test sequentially whether $s_z \in L$ for $z = 0, 1, 2, \ldots$ 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:

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

In the sequel we deal with the learnability of indexed families of uniformly recursive languages defined as follows (cf. Angluin, 1980b). A sequence $L_0, L_1, L_2, \ldots$ is said to be an *indexed family* $\mathcal{L}$ of uniformly recursive languages provided all L_j are non-empty 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 all what follows we refer to indexed families of uniformly recursive languages as indexed families for short. Moreover, we often denote an indexed family and its range by the same symbol $\mathcal{L}$. The meaning will be clear from the context.

As in Gold (1967) 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 space of hypotheses an enumerable family of grammars $G_0, G_1, G_2, \ldots$ over the terminal alphabet

Σ satisfying $\mathcal{L} \subseteq \{L(G_j) \,|\, j \in \mathbb{N}\}$. Moreover, we require that membership in $L(G_j)$ is uniformly decidable for all $j \in \mathbb{N}$ and all strings $s \in \Sigma^*$. The numbers j that the IIM outputs are then interpreted as $L(G_j)$. Moreover, for notational convenience we use $L(\mathcal{G})$ to denote $\{L(G_j) \,|\, j \in \mathbb{N}\}$ for every hypothesis space $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$.

A sequence $(j_x)_{x \in \mathbb{N}}$ of numbers is said to be convergent in the limit iff there is a number j such that $j_x = j$ for almost all numbers x. Now we define some concepts of learning. We start with learning in the limit.

Definition 1. **(Gold, 1967)** *Let $\mathcal{L}$ be an indexed family of languages, $L \in \mathcal{L}$, and let $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ be a space of hypotheses. An IIM M LIM–identifies L from a text t with respect to $\mathcal{G}$ iff it almost always outputs a hypothesis and the sequence $(M(t_x))_{x \in \mathbb{N}}$ converges in the limit to a number j such that $L = L(G_j)$.*

Furthermore, M LIM–identifies L iff M LIM–identifies L from every text $t \in text(L)$. We set:
$LIM(M) = \{L \in \mathcal{L} \,|\, M \; LIM\text{–identifies } L\}$.

Finally, let LIM denote the collection of all indexed families $\mathcal{L}$ for which there is an IIM M such that $\mathcal{L} \subseteq LIM(M)$.

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.

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, (1980))** *An IIM is said to be set-driven 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, $\hat{t}$ provided $t_x^+ = \hat{t}_y^+$.*

Schäfer (1984) as well as Fulk (1990), later, and independently proved that set-driven IIMs are less powerful than unrestricted ones. Fulk (1990) 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 priorily 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 (1986) 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 (1984), Osherson et al. (1986))** *An IIM is said to be rearrangement-independent 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, $\hat{t}$ provided $t_x^+ = \hat{t}_x^+$.*

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, 1967)** *Let $\mathcal{L}$ be an indexed family of languages, $L \in \mathcal{L}$ and let $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ be a space of hypotheses. An IIM M FIN–identifies L from text t with respect to $\mathcal{G}$ iff it outputs only a single and correct hypothesis j, i.e., $L = L(G_j)$, and stops.*

Furthermore, M FIN–identifies L, iff M FIN–identifies L from every $t \in text(L)$. We set: $FIN(M) = \{L \in \mathcal{L} \mid M\ FIN\text{–identifies } L\}$ and define the resulting learning type FIN to be the collection of all finitely inferable indexed families.

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 (1991), Wiehagen (1991))** *Let $\mathcal{L}$ be an indexed family of languages, $L \in \mathcal{L}$ and let $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ be a space of hypotheses. An IIM M is said to identify a language L from text*

(A) *strong-monotonically*

(B) *monotonically*

(C) *weak-monotonically*

iff

M LIM–identifies L 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} for some $k \geq 1, k \in N$, 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 $SMON$, MON, and $WMON$, we denote the family of all sets $\mathcal{L}$ of indexed families for which there is an IIM inferring it strong-monotonically, monotonically, and weak-monotonically, respectively.

Definition 6. **(Angluin, 1980b)** *Let $\mathcal{L}$ be an indexed family, $L \in \mathcal{L}$, and let $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ be a space of hypotheses. An IIM M $CONSERVATIVE$–identifies L from text with respect to $\mathcal{G}$ iff for every text t the following conditions are satisfied:*

(1) $L \in LIM(M)$ *w.r.t.* $\mathcal{G}$,

(2) *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 $t^+_{x+k} \not\subseteq L(G_{j_x})$.*

CONSERVATIVE(M) as well as the collections of sets $CONSERVATIVE$ are defined in an analogous manner as above. Note that $WMON = CONSERVATIVE$ (cf. Lange and Zeugmann (1993a)). Additionally, every conservatively working IIM satisfies the monotonicity constraints a weak-monotonically working IIM has to fulfill.

We conclude this subsection with the following convention. For any model of inference defined above, the prefix E denotes the requirement to learn an indexed family $\mathcal{L}$ with respect to the hypothesis space $\mathcal{L}$ (*exact* learning). Furthermore, class comprising inference is denoted by the prefix C, e.g. $CLIM$ denotes the set of all indexed families $\mathcal{L}$ inferable with respect to some hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ such that $\mathcal{L} \subseteq \{L(G_j) \,|\, j \in \mathbb{N}\}$. If no prefix is used, then class preserving learning is meant, i.e., inference with respect to a hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ such that $range(\mathcal{L}) = \{L(G_j) \,|\, j \in \mathbb{N}\}$. For example, $r-MON$ denotes the collection of all indexed families $\mathcal{L}$ that may be monotonically identified by a rearrangement-independent IIM with respect to some class preserving hypothesis space.

3. Learning with Set-driven IIMs.

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

Proof. $EFIN = FIN = CFIN$ is due to Lange and Zeugmann (1993b). 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, 1992) it follows that there exists a recursively generable family $(T_j)_{j\in\mathbb{N}}$ of finite non-empty 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 a IIM M witnessing $\mathcal{L} \in s-EFIN$. Let $L \in \mathcal{L}$, $t \in text(L)$, and $x \in \mathbb{N}$.

$M(t_x) =$ " If $x = 0$ or M when fed successively with 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, request the next input and output nothing."

It remains to show that $\mathcal{L} \subseteq s-EFIN(M)$. By construction, M uses $\mathcal{L}$ as hypothesis space.

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

Let $t \in text(L)$. It remains 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 $t^+ = L$, there must be an x such that $T_k \subseteq t_x^+$. Now, there is only one case imaginable that could prevent M to stop. Namely, it 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 the claim.

Claim 2. M is set-driven.

Let t_x and $\hat{t}_y$ be two initial text segments of a language $L \in \mathcal{L}$ such that $t_x^+ = \hat{t}_y^+$. We have to show that $M(t_x) = M(\hat{t}_y)$. Suppose, M executes stage x or y, respectively. Since $L \in \mathcal{L}$, there exists a least number k such that $L = L_k$. Hence, M finds indices i, j 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.

Since j is the least index with $t_z^+ \subseteq L_j$ and since $L_j = L$, we conclude that j is the smallest number such that $\hat{t}_y^+ \subseteq L_j$. Hence, $T_j \subseteq t_z^+ \subseteq \hat{t}_y^+$, and M outputs j.

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

Suppose $M(\hat{t}_m) = k$ for some $m < y$. As above, then $L_k = L$. However, as we already seen in the proof of Claim 1, M always outputs the least index of the language it actually learns. Hence, we obtain $k = j$. This proves the claim.

q.e.d.

As we have already mentioned, the examples of Schäfer (1984) and Fulk (1990) 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 (1993b). 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 $L_{\langle k,0\rangle} = \{a^k b^n \,|\, n \in \mathbb{N}^+\}$. 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:

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

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

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

Since the halting problem is undecidable, Claim A follows by contraposition of the following Claim B.

Claim B. If there exists an IIM M with $\mathcal{L} \subseteq s{-}CLIM(M)$, 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 w.r.t. some hypothesis space $\mathcal{G}$ comprising $\mathcal{L}$. 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 lexicographically ordered text t of $L_{\langle k,0\rangle}$ until M on input t_z outputs for the first time a hypothesis j such that $t_z^+ \cup \{a^k b^{z+1}\} \subseteq L(G_j)$.

(A2) Test whether $\Phi_k(k) \leq z$. In case it is, output "$\varphi_k(k)$ converges."
Otherwise output "$\varphi_k(k)$ diverges."

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 $t_z^+ \cup \{a^k b^{z+1}\} \subseteq L(G_j)$. Moreover, the test whether or not $t_z^+ \cup \{a^k b^{z+1}\} \subseteq L(G_j)$ 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 $\neg\ \Phi_k(k) \leq z$. Suppose the converse; then there exists a $y > z$ 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 $M(\hat{t}_{x+r}) = j$ for all $r \in \mathbb{N}$. Therefore, M fails to infer L on its text $\hat{t}$. This contradicts our assumption that $\mathcal{L} \subseteq s\text{-}CLIM(M)$. Hence, Claim B is proved.

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

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

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

(A) 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 (B).

(B) 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."

q.e.d.

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.

Theorem 3. *There is an indexed family $\mathcal{L}$ such that*

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

(2) $\mathcal{L} \not\subseteq LIM(M)$ *for all IIM M, provided M is set-driven,*

(3) *there is an IIM M such that* $\mathcal{L} \subseteq s\text{-}CSMON(M)$.

Proof. The desired indexed family $\mathcal{L} = (L_{\langle k,j\rangle})_{k,j\in N}$ is defined as follows. For all $k \in N$ we set $L_{\langle k,0\rangle} = \{a^k b^m \mid m \in N^+\}$. 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 $\mathcal{L} \in ESMON$. This is done as follows: Let $L \in \mathcal{L}$, $t \in$ *text*(L), and $x \in \mathbb{N}$.

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

(A) Test whether $\neg\ \Phi_k(k) \leq x$. 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 $\mathcal{L} \subseteq r{-}ESMON(M)$ 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 $j \in N$. 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 works strong-monotonically.

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

Suppose, $L = \{a^k b\}$. Since $t \in$ *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 $\langle k, 0\rangle$ afterwards. Clearly, this mind change fulfills the strong-monotonicity constraint. Hence, $\mathcal{L} \subseteq r{-}ESMON(M)$.

Claim 2. $\mathcal{L} \not\subseteq LIM(M)$ for all IIM M, provided M is set-driven.

This claim is proved via the following lemma.

Lemma 1. *Let M be any set-driven IIM such that $\mathcal{L} \subseteq LIM(M)$ w.r.t. some class preserving hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$. Then M may be used to decide the halting problem.*

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

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

(A1) 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 (A2).

(A2) 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."

Obviously, $\mathcal{A}$ is an algorithm. It remains to show that $\mathcal{A}$ behaves correctly. Suppose, $\mathcal{A}$ outputs "$\varphi_k(k) \uparrow$" but "$\varphi_k(k) \downarrow$." By definition of $\mathcal{L}$, $L = \{a^k b\} \in \mathcal{L}$. Since M is supposed to be *set-driven*, it has to output a correct hypothesis after having seen $a^k b$. But it does not, since $\mathcal{A}$ has terminated with "$\varphi_k(k) \uparrow$." This contradiction directly yields that $\mathcal{A}$ behaves correctly if it outputs "$\varphi_k(k) \uparrow$".

Now, let us assume $\mathcal{A}$ terminates with "$\varphi_k(k) \downarrow$." Taking into account that $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ is a class preserving hypothesis space, we directly see that $a^k b^2 \notin L(G_z)$ implies $L(G_z) = \{a^k b\}$. Hence, "$\varphi_k(k) \downarrow$." This proves the lemma.

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

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

First of all we define the desired class comprising hypothesis space. For all $k \in \mathbb{N}$ 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 $\mathcal{L} \in CSMON$ 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.
q.e.d.

Theorem 3 directly yields the following corollary which relates the power of set-driven and unrestricted IIMs to one another.

Corollary 4. For all $ID \in \{SMON, MON, WMON\}$:

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

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

Proof. This corrollary follows immediately from Theorem 3. There, we have shown $r{-}ESMON \setminus s-LIM \neq \emptyset$. This yields all the proper inclusions mentioned, since $ESMON \subset SMON$, $ESMON \subset EID \subset ID$ for all $ID \in \{MON, WMON\}$ (cf. Lange and Zeugmann (1993b)) as well as $s{-}EID \subseteq s{-}ID \subseteq s{-}LIM$ for all $ID \in \{SMON, MON, WMON\}$ by definition.
q.e.d.

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 looks differently 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 5. s-$CCONSERVATIVE = CCONSERVATIVE$

Proof. We partition the proof into two parts. First, we show that every indexed family in *CCONSERVATIVE* belongs to r-*CCONSERVATIVE* (cf. Lemma 1) below. Then we apply this result and show that set-drivenness does not restrict the power of class comprising conservative learning (cf. Lemma 2).

Lemma 1. *r–CCONSERVATIVE = CCONSERVATIVE*

Let $\mathcal{L} \in$ *CCONSERVATIVE*. By Theorem 14 in Lange and Zeugmann (1993c) there exists 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 non-empty sets such that

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

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

(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 non-empty sets that allows the design of a rearrangement-independent IIM inferring $\mathcal{L}$ conservatively w.r.t. $\mathcal{G}$. 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 (1) through (3), too. Next, we even show the following stronger result.

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

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

$M(t_x) =$ "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 works conservatively.

Let k and j be two hypotheses produced by M on input t_x and t_{x+r}, respectively. We have to show that $t_{x+r}^+ \not\subseteq L(G_k)$. 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, the claim is proved.

Claim 2. M infers L from t.

Let $z = \mu k\, [L(G_k) = L]$. Therefore, $L(G_j) \neq L$ for all $j \leq z$. 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, the claim follows and Statement 1 is proved.

Finally, since $\mathcal{L} \subseteq L(\mathcal{G})$, we may easily conclude that $\mathcal{L} \subseteq r-CCONSERVATIVE(M)$. This proves Lemma 1.

Lemma 2. Let $\mathcal{L}$ be any indexed family. If $\mathcal{L} \in$ *CCONSERVATIVE*, then there exists a hypothesis space $\tilde{\mathcal{G}} = (\tilde{G}_j)_{j\in\mathbb{N}}$ and an IIM $\tilde{M}$ such that $\mathcal{L} \subseteq s-CCONSERVATIVE(\tilde{M})$ w.r.t. $\tilde{\mathcal{G}}$.

First, we define the hypothesis space $\tilde{\mathcal{G}} = (\tilde{G}_j)_{j\in\mathbb{N}}$ as follows. Applying Lemma 1, there exists a hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ and and IIM such that $\mathcal{L} \subseteq r-CCONSERVATIVE(M)$ w.r.t. $\mathcal{G}$ as well as $L(\mathcal{G}) \subseteq r-ECONSERVATIVE(M)$. Afterwards, we use the latter statement and show a more general result which turns out to be quite helpful in order to prove Corollary 6. 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 1. 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 $rf(t_{x+1}) = rf(t_x), s_{x+1}$ otherwise. Obviously, given any initial segment t_x of a text t one can effectively compute $rf(t_x)$.

Statement 2. $L(\mathcal{G}) \in s-CCONSERVATIVE$

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

$\tilde{M}(t_x) =$ "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 1. $\tilde{M}$ is set-driven.

Let t_x and $\hat{t}_y$ be be two initial text segments of a language $L \in L(\mathcal{G})$ 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 works 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 the claim.

Claim 2. $\tilde{M}$ works conservatively.

By construction, $\tilde{M}$ outputs on any 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 works conservatively, 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, $\tilde{M}$ works conservatively.

Claim 3. $\tilde{M}$ 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 works conservatively. 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, it infers L.

Therefore, Statement 2 is proved. Since $\mathcal{L} \subseteq L(\mathcal{G})$, it follows $\mathcal{L} \subseteq \mathit{CCONSERVATIVE}(\tilde{M})$.
q.e.d.

Corollary 6. Let $\mathcal{L} \in \mathit{CCONSERVATIVE}$. Then, there is a hypothesis space $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ comprising $\mathcal{L}$ such that $L(\mathcal{G}) \in s-\mathit{ECONSERVATIVE}$.

Proof. Let $\mathcal{L} \in \mathit{CCONSERVATIVE}$. Furthermore, due to the latter theorem, there is an IIM $\tilde{M}$ and a hypothesis space $\tilde{\mathcal{G}}$ such that $\mathcal{L} \subseteq s-\mathit{CCONSERVATIVE}(\tilde{M})$ w.r.t. $\tilde{\mathcal{G}}$. Let $\tilde{M}$ and $\tilde{\mathcal{G}}$ be defined as in Lemma 2.

Recall that $\tilde{\mathcal{G}}$ is a canonical enumeration of grammars defining all languages in $\mathcal{L}$ 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 $L(\tilde{G}_j) \in s-\mathit{CCONSERVATIVE}(M)$. Otherwise, $L(\tilde{G}_j)$ is a finite language.

We start with the definition of the desired hypothesis space $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$. If j is even, then we set $G_j = \tilde{G}_j$. 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 $G_j = \tilde{G}_j$. In case it does not, we set $G_j = \tilde{G}_{j-1}$.

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

$M(t_x) =$ "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 $\tilde{M}(t_x) = j$. Output j and request the next input."

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

Now, let us suppose, $L \neq L(G_{2k})$ for some $k \in \mathbb{N}$. By definition of $\mathcal{G}$, 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 $\mathcal{G}$, and by assumption, we obtain the following. There is a number j such that $\tilde{M}(t_x) = j$, $L = t_x^+ = L(\tilde{G}_j) = L(G_j)$. Hence, $M(t_x) = j$, too. Finally, since M is set-driven, we directly get $M(t_y) = j$ for all $y \geq j$. Consequently, M learns L.
q.e.d.

The next theorem gives some more evidence that set-drivenness is not that restrictive as it might seem.

Theorem 7.

(1) $s-\mathit{SMON} \setminus \mathit{EWMON} \neq \emptyset$,

(2) $s-\mathit{CSMON} \setminus \mathit{WMON} \neq \emptyset$,

(3) $s-\mathit{EWMON} \setminus \mathit{MON} \neq \emptyset$.

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 $L_{\langle k,0\rangle} = \{a^k b^n \,|\, n \in \mathbb{N}^+\}$. 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: $L_{\langle k,j\rangle} = \{a^k b^m \,|\, 1 \le m \le \Phi_k(k)\}$.

In Lange and Zeugmann (1993b) 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 A. $\mathcal{L}_{sm} \in SMON$.

We have to show that there are a space of hypotheses $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ with $range(\mathcal{L}_{sm}) = L(\mathcal{G})$ and a set-driven IIM M such that M does strong-monotonically infer $\mathcal{L}$ w.r.t. $\mathcal{G}$.

First of all, we define the space of hypotheses $\mathcal{G}$. 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 $range(\mathcal{L}_{sm}) = L(\mathcal{G})$.

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

$M(t_x) =$ "Determine the unique k such that $t_0 = a^k b^m$ for some $m \in \mathbb{N}$. Test whether or not $t_x^+ \in L(G_{2k})$. 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 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 A, 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 $L_{\langle k,0\rangle} = \{a^k b^n \,|\, n \in \mathbb{N}^+\}$. For all $k \in \mathbb{N}$ and all $j \in \mathbb{N}^+$ we distinguish the following cases:

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

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

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

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

$L_{\langle k,j\rangle} = \{a^k b^m \,|\, 1 \le m \le \Phi_k(k)\} \cup \{a^k b^{\Phi_k(k)+2(d+m)} \,|\, m \in \mathbb{N}^+\}$

By reducing the halting problem to $\mathcal{L}_{csm} \in WMON$, one may prove that $\mathcal{L}_{csm} \notin WMON$. An IIM M witnessing $\mathcal{L}_{csm} \in s-CSMON$ can be easily designed, if one choose the following space of hypotheses $\mathcal{G} = (G_{\langle k,j\rangle})_{j,k\in\mathbb{N}}$. 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.

The remaining part can be easily shown. One has simply to choose the same indexed family as used in Lange and Zeugmann (1993a) in order to separate $WMON$ and MON.

q.e.d.

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 (1980b) characterized the learnability of those indexed families $\mathcal{L}$ that are inferable w.r.t. the hypothesis space $\mathcal{L}$ in terms of finite, and recursively enumerable tell-tales. Actually, she proved the slightly stronger result that $r\!-\!ELIM = ELIM$. Recently, we showed $ELIM = LIM = CLIM$ (cf. Lange and Zeugmann (1993d)), 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 (1984), and later, but independently by Fulk (1990) 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's (1984) and Fulk's (1990) is much stronger than Angluin's (1980b), since it is not restricted to the learnability of indexed families. By the next theorem we summarize the known results.

Theorem 8. (Angluin (1980b), Schäfer (1984), Fulk (1990))

$r\!-\!ELIM = ELIM = LIM = CLIM$

However, neither Schäfer's (1984) nor Fulk's (1990) 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 9.

(1) $r\!-\!ESMON = ESMON$,

(2) $r\!-\!SMON = SMON$.

Proof. First, we prove assertion (2).

Let $\mathcal{L} \in SMON$. Applying the characterization theorem for $SMON$ (cf. Lange and Zeugmann (1992)), we know that there exists a class preserving space of hypothesis $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ as well as a recursively generable family $(T_j)_{j\in\mathbb{N}}$ of finite non-empty 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 a IIM M witnessing $\mathcal{L} \in r\!-\!SMON$. So let $L \in \mathcal{L}$, $t \in text(L)$, and $x \in \mathbb{N}$.

$M(t_x) =$ "Search for the least $j \leq x$ for which $T_k \subseteq t_x^+ \subseteq L(G_k)$. 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 $\mathcal{L} \in SMON(M)$ w.r.t. the hypothesis space $\mathcal{G}$.

Claim 1. M infers L on text t.

Let $j = \mu z[L(G_z) = L]$. Hence, there is a least x such that $T_j \subseteq t_x^+$. Therefore, M will output ones a hypothesis. For all $k < j$ with $T_k \subseteq L$ we may conclude that $L(G_k) \subset L$. Otherwise, we obtain $L(G_j) = L(G_k) = L$, because of $T_k \subseteq L(G_k)$ and $T_j \subseteq L(G_j)$ (cf. (ii)). Hence, there exists a y such that $t_y^+ \not\subseteq L(G_k)$ for all $k < j$ with $T_k \subseteq L$. Therefore, $M(t_{y+r}) = j$ for all $r \in \mathbb{N}$. This proves the claim.

Claim 2. M works strong-monotonically.

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 $T_j \subseteq t_x^+ \subseteq L(G_k)$. Therefore, $L(G_j) \subseteq L(G_k)$ (cf. (ii)). This proves the claim.

To sum up, M is witnessing $\mathcal{L} \in r{-}SMON$. 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 exists a rearrangement-independent IIM $\hat{M}$ as well as a class preserving hypothesis space $\mathcal{G}$ such that $\mathcal{L} \subseteq r{-}SMON(M)$ w.r.t. the hypothesis space $\mathcal{G}$.

Applying Theorem 4 of Lange and Zeugmann (1993b), we know that there exists some total recursive function $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ satisfying

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

(ii) 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 $L \in \mathcal{L}$, $t \in \textit{text}(L)$, and $x \in \mathbb{N}$.

$M(t_x) =$ "Simulate $\hat{M}$ on input t_x. If $\hat{M}$ when fed successively 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 (A1). Otherwise, output nothing and request the next input.

(A1) 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 (A1) can be effectively accomplished, too. Furthermore, by property (i) of f and since $t_x^+ \subseteq L(G_j)$, instruction (A1) has to terminate for every $j \in \mathbb{N}$. Hence, M is indeed an IIM. Due to its definition, M is 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 (i) 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 successively hypotheses 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 (ii) guarantees $L_{f(j,y)} \subseteq L_{f(j,z)}$.

Case 2. $j \neq k$

Since f satisfies (i) and (ii), 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 $\hat{M}$ works strong-monotonically on every text for every language $L \in \mathcal{L}$, 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 it works strong-monotonically. This proves the theorem.

q.e.d.

It remains open whether or not the above theorem extends to the class comprising case. Next we consider monotonic language learning. Now, the situation is completely different, since we have the following theorem.

Theorem 10.

(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 1. $r{-}EMON \setminus s{-}MON \neq \emptyset$

By Theorem 3 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 the lemma.

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 is theorem is proved.

Lemma 2. $EMON \setminus r{-}MON \neq \emptyset$

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 bit of 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 $L \in \mathcal{L}$ from any text t for L w.r.t. the hypothesis space $\mathcal{L}$ itself. So let us assume any $L \in \mathcal{L}$, any text t for L and any $x \in \mathbb{N}$.

$M(t_x) =$ "If $x = 0$ or M has not produced any hypothesis when successively fed t_{x-1}, then execute (A1). Otherwise, goto (A2).

(A1) 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.

(A2) Let $M(t_{x-1}) = j$. If $t_x^+ \subseteq L_j$, then repeat the hypothesis j and request the next input.
Otherwise, goto (A3).

(A3) 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.
Otherwise, goto (A4).

(A4) 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 $\mathcal{L} \subseteq EMON(M)$. Obviously, for every $k \in \mathbb{N}$, M identifies L_{4k} as well as L_{4k+1} on every text for the corresponding language. Thereby, M does not perform any mind change at all. Hence, M works monotonically on every $t \in \textit{text}(L_{4k}) \cup \textit{text}(L_{4k+1})$, $k \in \mathbb{N}$. 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 works monotonically. 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 works monotonically. Otherwise, $4k+1$ is M's first hypothesis. Now, it is easy to verify that M produce 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 works monotonically.

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

Then, a quite similar argumentation yields that M works monotonically. 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 works monotonically.

Therefore, we obtain $\mathcal{L} \subseteq EMON(M)$, and the claim is proved.

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

Suppose the converse, i.e., there is a class preserving hypothesis space $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ and an IIM M such that $\mathcal{L} \subseteq r{-}MON(M)$ w.r.t. $\mathcal{G}$.

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

Next to, we define the desired algorithm.

Algorithm $\mathcal{A}$: "On input k execute (A1) until (α1) or (α2) are fulfilled, respectively. Afterwards, execute (A2).

(A1) For all $x = 0, 1, \ldots$, execute in parallel (α1) and (α2) until one of them is successful.

(α1) Test whether $\Phi_k(k) \leq x$.

(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 $a^k b \in L(G_m)$ and $a^k c \in L(G_{\hat{m}})$.

(A2) 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 $\Phi_k(k) \leq x + y$.

(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 accomplished effectively. Furthermore, since M is an IIM as well as membership is uniformly decidable for $L(\mathcal{G})$, (α2) and (β2) can be accomplished effectively, too. Hence, $\mathcal{A}$ is indeed an algorithm.

First, we show that $\mathcal{A}$ terminates for all $k \in \mathbb{N}$. Let us assume that the execution of (A1) 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 the languages $\{a^k b\}$, $\{a^k c\}$ from its uniquely defined text, a contradiction. Applying the same arguments one can show that the execution of (A2) has to terminate, too. Hence, $\mathcal{A}$ terminates on every input $k \in \mathbb{N}$.

It remains to show that algorithm $\mathcal{A}$ works correctly. Obviously, if $\mathcal{A}$ stops its computation with "$\varphi_k(k)$ converges", then $\varphi_k(k)$ is indeed defined. Suppose that $\mathcal{A}$ 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 $\hat{t}^+_x \subseteq L(G_{\hat{m}})$. After a bit of reflection, one can easily verify that $L(G_m) = L_{4k}$ and $L(G_{\hat{m}}) = L_{4k+1}$. This is caused by the assumption that M works monotonically w.r.t. a class preserving hypothesis space. 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 $M(\hat{t}_{x+y}) = m$, where $\hat{t}_{x+y} = \underbrace{a^k c, \dots, a^k c}_{(x+1)-times} \underbrace{a^k b, \dots, a^k b}_{y-times}$, since $t^+_{x+y} = \hat{t}^+_{y+x}$. Because $\{a^k b, a^k c\} \subseteq L(G_n)$ as well as $L(G_n) \in \mathcal{L}$, it suffices to distinguish the following two cases.

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

Since $\mathcal{A}$ terminates with "$\varphi_k(k)$ diverges", we obtain $\Phi_k(k) > x + y$. Therefore, $\hat{t}_{x+y}$ is an initial segment of a text for L_{4k+3}. 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 $\mathcal{A}$. Thus, Claim 3 is shown.

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

q.e.d.

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 one might expect that rearrangement-independence is a severe restriction under the weak-monotonic constraint, too. On the other hand, looking at theorem 5 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 11. *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 $L \in \mathcal{L}$ 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 such that $\mathcal{L} \subseteq ECONSERVATIVE(M)$ w.r.t. $\mathcal{L}$. 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

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

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 $L \in \mathcal{L}$ 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 $w_0, ..., w_x$ outputs j, where $w_0, ..., w_x$ are the strings of T_j^y written in canonical order w.r.t. L_j. Furthermore, $T_j^y \subseteq L_z$ and therefore, $w_0, ..., w_x$ is an initial segment of a text for L_z. Since M has to infer L_z from every text, it has to perform at least one mind change on every text $t \in \mathit{text}(L_z)$ beginning with $w_0, ..., w_x$ that cannot be caused by an inconsistency. This contradiction proves (3).

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

$M(t_x) =$ "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 non-empty 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 (1992), it is easy to see that $\mathcal{L} \subseteq \mathit{ECONSERVATIVE}(M)$. We omit the details.

q.e.d.

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

Theorem 12.

(1) $r{-}\mathit{ECONSERVATIVE} = \mathit{ECONSERVATIVE}$,

(2) $r{-}\mathit{CONSERVATIVE} = \mathit{CONSERVATIVE}$.

Proof. First we prove assertion (1). Let $\mathcal{L} \in \mathit{ECONSERVATIVE}$. By Theorem 11 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 11 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 11. 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 $m = \mu y[\hat{T}_j^y \neq \emptyset]$ und $n = \mu y[\hat{T}_k^y \neq \emptyset]$. 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 the claim.

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

$M(t_x) =$ "Test for all $k \leq x$ whether or not $\hat{T}_k^x \neq \emptyset$. For all non-empty $\hat{T}_k^x$ 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 $\mathcal{L} \subseteq r-ECONSERVATIVE(M)$. Obviously, M is rearrangement-independent.

Claim 2. M works conservatively.

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 works conservatively, and the claim is proved.

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

Let $L \in \mathcal{L}$ and let $t \in \mathit{text}(L)$. 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 $k \in K$ be such that there is no $\hat{k}$ with $\hat{k} < k$ and $\mu y[\hat{T}_{\hat{k}}^y \neq \emptyset] \leq \mu y[\hat{T}_k^y \neq \emptyset]$. We show that M converges to k. By Claim 1, we conclude that $\hat{T}_{\hat{k}}^x \neq \emptyset$ and $\hat{T}_k^y \neq \emptyset$ implies $\hat{T}_{\hat{k}}^x \subseteq \hat{T}_k^y$ for all $\hat{k} \in K$ with $\hat{k} \neq k$. Therefore, if M outputs at least once a correct hypothesis for L, then this hypothesis is k. Let $y_k = \mu y[\hat{T}_k^y \neq \emptyset]$. Since t is a text for L, there exists a $x \geq y_k$ 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. Let $y_j = \mu y[\hat{T}_j^y \neq \emptyset]$, then C may written as follows.

$$C = \{j \mid j \in \mathbb{N},\ y_j \leq y_k,\ \hat{T}_j^{y_j} \subseteq t^+\}.$$

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 $C = \{j \mid j \leq y_k,\ y_j \leq y_k,\ \hat{T}_j^{y_j} \subseteq t^+\}$. Consequently, C is finite. Finally, applying property (3) of Theorem 11 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 $j \neq k$ there exists an x_j such that $t_{x_j} \not\subseteq L_j$. Since C is finite, it successively shrinks to $\{k\}$, and hence M converges to k.

This proves assertion (1) of the theorem. Assertion (2) can be proved analogously as Lemma 1 in the proof of Theorem 5.

q.e.d.

5. Summary

We start with the following figure that summarizes the results obtained and points to questions that remain open. We shall discuss them and some less obvious ones in this section.

	exact learning	class preserving learning	class comprising learning
FIN	*set drivenness* +	*set drivenness* +	*set drivenness* +
SMON	*rearrangement independence* +	*rearrangement independence* +	**?**
MON	*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 $s-ID \subset ID$. "*Rearrangement-independence* $-$" implies $s-ID \subset r-ID \subset ID$ whereas "*set-drivenness* $+$" should be interpreted as $s-ID = ID$ and, therefore, $r-ID = ID$, too.

Besides the two open problems we have pointed to in the figure above, there are two more intriguing questions deserving attention. 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 charcterizations 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 $\mathcal{P} \neq \mathcal{NP}$.

6. References

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

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

[3] Berwick, R. (1985), "The Acquisition of Syntactic Knowledge," Cambridge, Mass.: MIT Press.

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

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

[6] Hopcroft, J.E., and Ullman, J.D (1969), "Formal Languages and their Relation to Automata," Addison-Wesley, Reading, Massachusetts.

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

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

[8] Lange, S., and Zeugmann, T. (1992), Types of monotonic language learning and their characterization, *in* "Proceedings 5th Annual ACM Workshop on Computational Learning Theory," July 27 - 29, Pittsburgh, pp. 377 - 390, ACM Press.

[9] Lange, S., and Zeugmann, T. (1993a), Monotonic versus non-monotonic language learning, *in* "Proceedings 2nd International Workshop on Nonmonotonic and Inductive Logic," December 1991, Reinhardsbrunn, (G. Brewka, K.P. Jantke and P.H. Schmitt, Eds.), Lecture Notes in Artificial Intelligence 659, pp. 254 - 269, Springer-Verlag, Berlin.

[10] Lange, S., and Zeugmann, T. (1993b), Language learning in dependence on the space of hypotheses, *in* "Proceedings 6th Annual ACM Conference on Computational Learning Theory," Santa Cruz, July 1993, pp. 127 - 136, ACM Press, New York.

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

[12] Lange, S., and Zeugmann, T. (1993d), Learning recursive languages with bounded mind changes, *International Journal of Foundations of Computer Science*, to appear.

[13] Machtey, M., and Young, P. (1978) "An Introduction to the General Theory of Algorithms," North-Holland, New York.

[14] Osherson, D., Stob, M., and Weinstein, S. (1986), "Systems that Learn, An Introduction to Learning Theory for Cognitive and Computer Scientists," MIT-Press, Cambridge, Massachusetts.

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

[16] Wexler, K. (1992), The subset principle is an intensional principle, *in* "Knowledge and Language: Issues in Representation and Acquisition," (E. Reuland and W. Abraham, Eds.), Kluwer Academic Publishers.

[17] Wexler, K., and Culicover, P. (1980), "Formal Principles of Language Acquisition," Cambridge, Mass.: MIT Press.

[18] Wiehagen, R. (1991), A thesis in inductive inference, *in* "Proceedings First International Workshop on Nonmonotonic and Inductive Logic," December 1990, Karlsruhe, (J. Dix, K.P. Jantke and P.H. Schmitt, Eds.), Lecture Notes in Artificial Intelligence Vol. 543, pp. 184 - 207, Springer-Verlag, Berlin.