Incremental Learning from Positive Data *

Steffen Lange
HTWK Leipzig
FB Mathematik und Informatik
PF 66
04521 Leipzig, Germany
steffen@informatik.th-leipzig.de

Thomas Zeugmann
Research Institute of
Fundamental Information Science
Kyushu University 33
Fukuoka 812, Japan
thomas@i.kyushu-u.ac.jp

Abstract

The present paper deals with a systematic study of incremental learning algorithms. The general scenario is as follows. Let c be any concept; then every infinite sequence of elements exhausting c is called *positive presentation* of c. An algorithmic learner successively takes as input one element of a positive presentation as well as its previously made hypothesis at a time, and outputs a new hypothesis about the target concept. The sequence of hypotheses has to converge to a hypothesis correctly describing the concept to be learned. This basic scenario is referred to as *iterative learning*. Iterative inference can be refined by allowing the learner to store an *a priori* bounded number of carefully chosen examples resulting in *bounded example memory inference* (cf. Fulk, Jain and Osherson [7]). Additionally, *feed-back* identification is introduced. Now, the learner is enabled to ask whether or not a particular element did already appear in the data provided so far.

Our results are threefold. First, the learning capabilities of the various models of incremental learning are related to previously studied learning models. It is proved that incremental learning can be always simulated by inference devices that are both set-driven and conservative. Second, feed-back learning is shown to be more powerful than iterative inference, and its learning power is incomparable to that of bounded example memory inference which itself extends that of iterative learning, too. In particular, the learning power of bounded example memory inference always increases if the number of examples the learner is allowed to store is incremented.

Third, a sufficient condition for iterative inference allowing *non-enumerative* learning is provided. The results obtained provide strong evidence that there is no unique way to design superior incremental learning algorithms. Instead, incremental learning is the art of knowing what to overlook.

Key words: Algorithmic Learning Theory, Computational issues in A.I., Concept learning

*This is a substantially revised version of the authors' paper at the Eighth Australian Joint Conference on Artificial Intelligence (AI'95), Canberra, November 1995.

1. INTRODUCTION

One of the main topics in cognitive science, epistemology, linguistic and psycholinguistic theory as well as of machine learning and algorithmic learning theory is language acquisition. The human ability to acquire their mother tongue as well as other languages has attracted a huge amount of interest in all these scientific disciplines. In particular, the main goal of the research undertaken is to gain a better understanding of what learning really is. Human language learning can be also considered as a an important example of incremental learning. However, the human ability to learn is by no means restricted to languages. Therefore, we consider in the present paper general systems that map evidence on a concept into hypotheses about it. We deal with scenarios in which the sequence of hypotheses *stabilizes* to an *accurate* and *finite* description of the target concept. Consequently, after having seen only finitely many data of the possibly infinite target, the algorithmic device performing the mapping of the data to hypotheses reaches its (generally unknown) point of convergence to a correct and finite description of the target concept. Clearly, then some form of learning must have taken place. Formalizing the notions "evidence," "stabilization," and "accuracy" results in the model of learning in the limit introduced by Gold [11]. During the last three decades much has been learned about the classes of formal languages and partial recursive functions that can successfully learned within Gold's [11] model and variations thereof (cf., e.g., [2, 5, 7, 8, 18, 24, 25, 27, 31]). We continue along these lines of research. In particular, we aim to investigate the principal learning capabilities of learners which perform incremental learning.

For the purpose of motivation and discussion of our research, we introduce some notations. A *positive presentation* of a concept c is an infinite sequence of elements that eventually exhausts all and only the elements of c. An algorithmic learner, henceforth called *inductive inference machine* (briefly IIM), takes as input initial segments of a positive presentation, and outputs, from time to time, a hypothesis about the target concept. The set $\mathcal{H}$ of all admissible hypotheses is called *hypothesis space*. Furthermore, the sequence of hypotheses has to converge to a hypothesis correctly describing the concept to be learned, i.e., after some point, the IIM stabilizes to an accurate hypothesis. If there is an IIM that learns a concept c from all positive presentations for it, then c is said to be *learnable in the limit* with respect to the hypothesis space $\mathcal{H}$ (cf. Definition 1).

However, this model makes the unrealistic assumption that the learner has access to the whole initial segment of a positive presentation provided so far. Clearly, each practical learning system has to deal with the limitations of space. Therefore, we investigate variations of the general approach described above that restrict the accessibility of input data. We deal with *iterative* learning, *bounded example memory* inference, and *feed-back* identification (cf. Definitions 3, 4, 5). All these models formalize *incremental learning*, a topic attracting more and more attention in the machine learning community (cf., e.g., [6, 10, 19, 22]). An iterative learner is required to produce its actual guess exclusively from its previous one and the next element in the positive presentation. Iterative learning has been introduced by Wiehagen [26] who studied it in the setting of learning recursive functions. Further results concerning this learning model can be found in [7, 8, 13, 14, 18, 19, 25, 26, 29, 31]. Osherson *et al.* [18] also considered the variant that the learner has access to the *last* k elements, where k is *a priori* fixed. Interestingly enough, the latter approach does *not* increase the learning power. Alternatively, Fulk *et al.* [7] considered learners that are allowed to store k *carefully chosen* examples, where k is again *a priori* fixed (bounded example memory inference). They proved bounded example memory learning to be more powerful than iterative inference, and stated, without proof, an *infinite* hierarchy of more and more powerful learners in dependence on the number of examples the learner is allowed to store. We provide a proof for this hierarchy (cf. Theorem 6). This result provides strong evidence that learning is the art of *knowing what to overlook*. Bounded example memory inference of particular concept classes has been also studied by Ameur [1] within Angluin's [3] on-line learning model.

Furthermore, we introduce feed-back identification for learning from positive data (cf. Definition 5). In this setting, the iterative learner is additionally allowed to ask whether or not a particular element did already appear in the data provided so far. Again, the learning capability considerably increases but the supplementary learning power is incomparable to those of bounded example memory inference (cf. Theorems 6 and 10). In particular, feed-back learners can simulate finite identification from positive and negative data (cf. Definition 7) while bounded example memory ones cannot (cf. Theorem 9 and Corollary 8). The latter result provides strong evidence that there is no unique way to design superior space efficient inference procedures, and leads to the problem to what extent the learning capabilities of iterative learning are extended. We answer this question by showing that all models of incremental learning are less powerful than inference devices that are both conservative and set-driven (cf. Theorem 5 and Proposition 1).

Finally, we provide a sufficient condition for iterative inference allowing *non-enumerative* learning (cf. Theorem 12). This condition is a natural sharpening of Angluin's [2] *finite thickness* (cf. Definition 9).

The paper is structured as follows. Section 2 presents notations and formal definitions of the learning models studied throughout this paper. All results are presented in Section 3. The limitations of incremental learning are established in Subsection 3.1 while Subsection 3.2 is dealing with their strength. In Subsection 3.3 we provide the announced sufficient condition for iterative learning. Finally, we discuss the results obtained and outline open problems.

2. FORMALIZING INCREMENTAL LEARNING

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 ***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. Blum [4]).

By $\langle\cdot,\cdot\rangle : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ we denote ***Cantor's pairing function***. Moreover, we use π_1 and π_2 to denote the ***projection functions*** over $\mathbb{N} \times \mathbb{N}$ to the first and second component, respectively. That is, $\pi_1\langle x, y\rangle = x$ and $\pi_2\langle x, y\rangle = y$ for all $x, y \in \mathbb{N}$.

Any recursively enumerable set $\mathcal{X}$ is called a ***learning domain***. By $\wp(\mathcal{X})$ we denote the power set of $\mathcal{X}$. Let $\mathcal{C} \subseteq \wp(\mathcal{X})$, and let $c \in \mathcal{C}$; then we refer to $\mathcal{C}$ and c as to a ***concept***

class and a ***concept***, respectively. Let c be a concept, and let $t = x_0, x_1, x_2, ...$ an infinite sequence of elements from c such that range$(t) = \{x_k \mid k \in \mathbb{N}\} = c$. Then t is said to be a ***positive presentation*** or, synonymously, a ***text*** for c. By $pos(c)$ we denote the set of all positive presentations for c. Moreover, let t be a positive presentation, and let y be a number. Then, t_y denotes the initial segment of t of length $y+1$, and $t_y^+ =_{df} \{x_k \mid k \leq y\}$. Furthermore, let $\sigma = x_0, \ldots, x_n$ be any finite sequence. Then we use $|\sigma|$ to denote the ***length*** of σ. Additionally, we use $\sigma \cdot t$ to denote the positive presentation obtained by concatenating σ and t provided $\sigma^+ \subseteq$ range(t).

In the sequel we deal with the learnability of indexable concept classes with uniformly decidable membership defined as follows (cf. Angluin [2]). A class of non-empty concepts $\mathcal{C}$ is said to be an ***indexable class*** with uniformly decidable membership provided there are an effective enumeration $c_0, c_1, c_2, ...$ of all and only the concepts in $\mathcal{C}$ and a recursive function f such that for all $j \in \mathbb{N}$ and all elements $x \in \mathcal{X}$ we have

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

In the following we refer to indexable classes with uniformly decidable membership as to indexable classes for short. Next, we describe some well-known examples of indexable classes. Let Σ denote any fixed finite alphabet of symbols, and let Σ^* be the free monoid over Σ. We set $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$, where ε denotes the empty string. Then $\mathcal{X} = \Sigma^*$ serves as the learning domain. As usual, we refer to subsets $L \subseteq \Sigma^*$ as to languages (instead of concepts). Then, the set of all context sensitive languages, context free languages, regular languages, and pattern languages, respectively, form indexable classes (cf. [12, 2]).

Next, let $X_n = \{0,1\}^n$ be the set of all n-bit Boolean vectors. We consider $\mathcal{X} = \bigcup_{n\geq 1} X_n$ as learning domain. Then, the set of all concepts expressible as a monomial, a k-CNF, a k-DNF, and a k-decision list form indexable classes (cf. [23, 20]).

As in Gold [11] we define an ***inductive inference machine*** (abbr. IIM) to be an algorithmic device working as follows: The IIM takes as its input larger and larger initial segments of a positive presentation and it either requests the next input element, or it first outputs a hypothesis, i.e., a number encoding a certain computer program, and then it requests the next input element.

The indices output by an IIM are interpreted with respect to a suitably chosen hypothesis space $\mathcal{H}$. Since we exclusively deal with indexable classes $\mathcal{C}$, we always take as a hypothesis space an indexable class $\mathcal{H} = (h_j)_{j\in\mathbb{N}}$. The indices are regarded as suitable finite encodings of the concepts described by the hypotheses. When an IIM outputs a number j, we interpret it to mean that the machine is hypothesizing h_j. Clearly, $\mathcal{H}$ must be defined over some learning domain $\mathcal{Z}$ comprising the learning domain $\mathcal{X}$ over which $\mathcal{C}$ is defined, and, moreover, $\mathcal{H}$ must *comprise* the target concept class $\mathcal{C}$. We say that a hypothesis $h \in \mathcal{H}$ describes a concept $c \in \mathcal{C}$ iff $c = h$, i.e., for all $z \in \mathcal{Z}$, $z \in h$ iff $z \in c$.

Let t be a positive presentation, and let $y \in \mathbb{N}$. Then we use $M(t_y)$ to denote the last hypothesis produced by M when successively fed t_y. The sequence $(M(t_y))_{y\in\mathbb{N}}$ is said to ***converge*** to the number j iff either $(M(t_y))_{y\in\mathbb{N}}$ is infinite and all but finitely many terms of it are equal to j, or $(M(t_y))_{y\in\mathbb{N}}$ is non-empty and finite, and its last term is j. Now we define some models of learning. We start with learning in the limit.

DEFINITION 1 (Gold [11]). *Let $\mathcal{C}$ be an indexable class, let c be a concept, and let $\mathcal{H} = (h_j)_{j \in \mathbb{N}}$ be a hypothesis space.* ***An IIM M LIM–identifies c from positive data with respect to $\mathcal{H}$*** *iff for every positive presentation t for c, there exists a $j \in \mathbb{N}$ such that the sequence $(M(t_y))_{y \in \mathbb{N}}$ converges to j and $c = h_j$.*

Furthermore, M LIM–identifies $\mathcal{C}$ with respect to $\mathcal{H}$ iff, for each $c \in \mathcal{C}$, M LIM–identifies c from positive data with respect to $\mathcal{H}$.

Finally, let LIM denote the collection of all indexable classes $\mathcal{C}$ for which there are an IIM M and a hypothesis space $\mathcal{H}$ such that M LIM–identifies $\mathcal{C}$ with respect to $\mathcal{H}$.

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

Within the next definition we consider the restriction that the IIM is never allowed to output hypotheses describing proper supersets of the target concept. Inductive inference machines behaving thus are called ***conservative***.

DEFINITION 2 (Angluin [2]). *Let $\mathcal{C}$ be an indexable class, let c be a concept, and let $\mathcal{H} = (h_j)_{j \in \mathbb{N}}$ be a hypothesis space.* ***An IIM M CONSV–identifies c from positive data with respect to $\mathcal{H}$*** *iff*

(1) *M LIM–identifies c from positive data with respect to $\mathcal{H}$,*
(2) *for every $t \in pos(c)$ and for all $y, k \in \mathbb{N}$, if $M(t_y) \neq M(t_{y+k})$ then $t^+_{y+k} \not\subseteq h_{M(t_y)}$.*

Finally, M CONSV–identifies $\mathcal{C}$ with respect to $\mathcal{H}$ iff, for each $c \in \mathcal{C}$, M CONSV–identifies c from positive data with respect to $\mathcal{H}$.

By CONSV we denote the collection of all indexable classes $\mathcal{C}$ for which there are an IIM M and a hypothesis space $\mathcal{H}$ such that M CONSV–identifies $\mathcal{C}$ with respect to $\mathcal{H}$.

Looking at the above definitions, we see that an IIM M has always access to the whole history of the learning process, i.e., in order to compute its actual guess M is fed all examples seen so far. In contrast to that, next we define ***iterative IIMs*** and a natural generalization of them called ***bounded example memory IIMs***. An iterative IIM is only allowed to use its last guess and the next element in the positive presentation of the target concept for computing its actual guess. Conceptionally, an iterative IIM M defines a sequence $(M_n)_{n \in N}$ of machines each of which takes as its input the output of its predecessor. Hence, the IIM M has always to produce a hypothesis.

DEFINITION 3 (Wiehagen [26]). *Let $\mathcal{C}$ be an indexable class, let c be a concept, and let $\mathcal{H} = (h_j)_{j \in \mathbb{N}}$ be a hypothesis space.* ***An IIM M IT–identifies c from positive data with respect to $\mathcal{H}$*** *iff for every positive presentation $t = (x_j)_{j \in \mathbb{N}}$ for c the following conditions are satisfied:*

(1) *for all $n \in \mathbb{N}$, $M_n(t)$ is defined, where $M_0(t) =_{df} M(x_0)$ and for all $n \geq 0$: $M_{n+1}(t) =_{df} M(M_n(t), x_{n+1})$,*
(2) *the sequence $(M_n(t))_{n \in \mathbb{N}}$ converges to a number j such that $c = h_j$.*

Finally, M IT–identifies $\mathcal{C}$ with respect to $\mathcal{H}$ iff, for each $c \in \mathcal{C}$, M IT–identifies c with respect to $\mathcal{H}$.

The resulting learning type IT is analogously defined as above.

In the latter definition $M_n(t)$ denotes the last, i.e., $(n+1)$th hypothesis output by M when successively fed the positive presentation t. Since M has to output a hypothesis in each learning step, it is justified to make the following convention. Let $\sigma = x_0, \dots, x_n$ be any finite sequence of elements over the relevant learning domain. Moreover, let $\mathcal{C}$ be any concept class over $\mathcal{X}$, and let M be any IIM that iteratively learns $\mathcal{C}$. Then we denote by $M_y(\sigma)$ the $(y+1)$th hypothesis output by M when successively fed σ provided $y \leq n$, and there exists a concept $c \in \mathcal{C}$ with $\sigma^+ \subseteq c$. We adopt this convention to the learning types defined below.

Within the following definition we consider a natural relaxation of iterative learning which we call bounded example memory inference. Now, an IIM M is allowed to memorize an *a priori* bounded number of the examples it already has had access to during the learning process. Again, M defines a sequence $(M_n)_{n\in\mathbb{N}}$ of machines each of which takes as input the output of its predecessor. Consequently, a bounded example memory IIM has to output a hypothesis as well as a subset of the set of examples seen so far.

Definition 4 (Fulk *et al.* [7]). *Let $k \in \mathbb{N} \cup \{*\}$, let $\mathcal{C}$ be an indexable class, let c be a concept, and let $\mathcal{H} = (h_j)_{j\in\mathbb{N}}$ be a hypothesis space.* ***An IIM M BEM$_k$–identifies c from positive data with respect to $\mathcal{H}$*** *iff for every positive presentation $t = (x_j)_{j\in\mathbb{N}}$ for c the following conditions are satisfied:*

(1) *for all $n \in \mathbb{N}$, $M_n(t)$ is defined, where $M_0(t) =_{df} M(x_0) = \langle j_0, S_0\rangle$ such that $S_0 \subseteq t_0^+$ and $\mathrm{card}(S_0) \leq k$, and for all $n \geq 0$: $M_{n+1}(t) =_{df} M(M_n(t), x_{n+1}) = \langle j_{n+1}, S_{n+1}\rangle$ such that $S_{n+1} \subseteq S_n \cup \{x_{n+1}\}$ and $\mathrm{card}(S_{n+1}) \leq k$,*
(note that $k = *$ means at most finitely many.)

(2) *the j_n in the sequence $(\langle j_n, S_n\rangle)_{n\in\mathbb{N}}$ of M's guesses converges to a number j such that $c = h_j$.*

Finally, M BEM$_k$–identifies $\mathcal{C}$ with respect to $\mathcal{H}$ iff, for each $c \in \mathcal{C}$, M BEM$_k$–identifies c with respect to $\mathcal{H}$.

For every $k \in \mathbb{N}$, the resulting learning type BEM_k is analogously defined as above. By definition, $IT = BEM_0$ as well as $BEM_* = LIM$.

Finally, we define learning by feed-back IIMs. The idea of feed-back learning goes back to Wiehagen [26] who considered it in the setting of inductive inference of recursive functions. However, his definition cannot be directly applied to learning from positive data. Informally, a feed-back IIM M is an iterative IIM that is additionally allowed to make a particular type of queries. In each learning Stage $n+1$, M has access to the actual input x_{n+1}, and its previous guess j_n. However, M is additionally allowed to compute a query from x_{n+1} and j_n. The query concerns the history of the learning process. That is, an element x and a "YES/NO" answer A are computed such that $A = 1$ iff $x \in t_n^+$ and $A = 0$, otherwise. Intuitively, M can just ask whether or not a particular string has already been presented in previous learning stages.

DEFINITION 5. *Let $\mathcal{C}$ be an indexable class, let c be a concept, and let $\mathcal{H} = (h_j)_{j\in\mathbb{N}}$ be a hypothesis space. Moreover, let $Q\colon \mathbb{N} \times \mathcal{X} \to \mathcal{X}$, and $A\colon \mathcal{X} \to \{0,1\}$ be computable total mappings.* ***An IIM M FB–identifies c from positive data with respect to $\mathcal{H}$*** *iff for every positive presentation $t = (x_j)_{j\in\mathbb{N}}$ for c the following conditions are satisfied:*

(1) *for all $n \in \mathbb{N}$, $M_n(t)$ is defined, where $M_0(t) =_{df} M(x_0)$ and for all $n \geq 0$: $M_{n+1}(t) =_{df} M(M_n(t), A(Q(M_n(t), x_{n+1})), x_{n+1})$,*

(2) *the sequence $(M_n(t))_{n\in\mathbb{N}}$ converges to a number j such that $c = h_j$ provided that A truthfully answers the questions computed by Q.*

Finally, M FB–identifies $\mathcal{C}$ with respect to $\mathcal{H}$ iff there are computable mappings Q and A as described above such that, for each $c \in \mathcal{C}$, M FB–identifies c with respect to $\mathcal{H}$.

3. RESULTS

In this section we relate the learning power of all the models introduced to one another. In particular, we deal with the limitations of all models of incremental learning by comparing their learning capabilities with conservative inference. Moreover, we provide results showing that rich concepts classes are incrementally learnable.

3.1. *On the Limitations of Incremental Learning*

All the models of incremental learning introduced above pose serious restrictions to the accessibility of data provided during the learning process. Therefore, one might readily expect a certain loss of learning power, i.e., $IT \subset LIM$, $FB \subset LIM$ as well as $BEM_k \subset LIM$ for all $k \in \mathbb{N}$. As far as iterative learning is concerned, this has been rigorously proved in Lange and Zeugmann [14]. Hence, the more interesting question is *how much* learning power is actually lost. Answering this question, we have to take into account that learnability has been defined with respect to *suitably chosen* hypothesis spaces. As pointed out in Lange and Zeugmann [15], conservative learning is sensitive with respect to the set of allowed hypothesis spaces and so is iterative learning (cf. [31]). Therefore, it is appropriate to illustrate this dependence which is done by our next theorem.

THEOREM 1. *There are an indexable class $\mathcal{C}$ and a hypothesis space $\mathcal{H}$ for it such that*

(1) *$\mathcal{C}$ is iteratively learnable with respect to $\mathcal{H}$, and*

(2) *no conservative IIM can infer $\mathcal{C}$ with respect to $\mathcal{H}$.*

Proof. We define the desired indexable class $\mathcal{C}$ via the following enumeration of languages $\mathcal{L} = (L_{\langle k,j\rangle})_{k,j\in\mathbb{N}}$. Without loss of generality, we may assume that $\Phi_k(k) \geq 1$ for all $k \in \mathbb{N}$. Now, let $k, j \in \mathbb{N}$; we distinguish the following cases.

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

Then, we set $L_{\langle k,j\rangle} = \{a^k b^\ell a^m \mid \ell, m \in \mathbb{N}^+\}$.

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

Set $d = j - \Phi_k(k) + 1$.

Subcase 2.1. $d \leq \Phi_k(k)$

Let $L_{\langle k,j\rangle} = \{a^k b^\ell a^m \mid 1 \leq \ell \leq d,\ m \in \mathbb{N}^+\}$.

Subcase 2.2. $d > \Phi_k(k)$

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

Since the predicate '$\Phi_i(x) \leq y$' is recursive in i, x, and y, membership is uniformly decidable with respect to the enumeration $\mathcal{L} = (L_{\langle k,j\rangle})_{k,j\in\mathbb{N}}$. We set $\mathcal{C} = \text{range}(\mathcal{L})$, and take $\mathcal{L}$ as the desired hypothesis space $\mathcal{H}$.

Claim 1. *There is an iterative IIM M learning $\mathcal{C}$ with respect to $\mathcal{L}$.*

Let $L \in \mathcal{C}$ and let $t = (s_j)_{j\in\mathbb{N}} \in pos(L)$. The IIM M is defined in stages, where Stage n conceptually describes M_n.

Stage 0. M receives as input s_0.

Determine the unique $k \in \mathbb{N}$ such that $s_0 = a^k b^\ell a^m$ for some $\ell, m \in \mathbb{N}^+$. Set $j_0 = \langle k, 0\rangle$, output j_0, and go to Stage 1.

Stage n, $n \geq 1$. M receives as input j_{n-1} and the $(n+1)$st element s_n of t.

Determine the $k, \ell, m \in \mathbb{N}$ such that $s_n = a^k b^\ell a^m$.

Case 1. $j_{n-1} = \langle k, 0\rangle$

Test whether or not $\Phi_k(k) \leq m$. In case it is, set $j_n = \langle k, \Phi_k(k)\rangle$. Otherwise, set $j_n = j_{n-1}$.

Case 2. $j_{n-1} = \langle k, z\rangle$ for some $z \in \mathbb{N}^+$

Determine $y = \Phi_k(k)$. Test whether or not $s_n = a^k b^\ell a^m \in L_{\langle k,z\rangle}$. In case it is, set $j_n = j_{n-1}$. Otherwise, set $j_n = \langle k, y + \ell - 1\rangle$.

Output j_n, and go to Stage $n + 1$.

By construction, M is an iterative IIM. We have to show that M infers $\mathcal{C}$. Let $k, z \in \mathbb{N}$, let $L = L_{\langle k,z\rangle}$, and let $t \in pos(L)$. If $\Phi_k(k)$ is not defined, then M outputs in every stage the guess $\langle k, 0\rangle$. By definition, $L_{\langle k,z\rangle} = L_{\langle k,0\rangle}$ for all $z \in \mathbb{N}$, and thus, $L = L_{\langle k,0\rangle}$.

Now assume $\Phi_k(k)$ to be defined. Hence there is a y such that $\Phi_k(k) = y$. By $\mathcal{C}$'s definition, L in particular contains all strings s of form $a^k b a^m$ with $m \geq y$. Since $t \in pos(L)$, M eventually receives a string $s = a^k b a^m$, $m \geq y$, and verifies $\Phi_k(k) \leq m$. Consequently, it rejects its initial guess $\langle k, 0\rangle$ and changes its mind to $\langle k, y\rangle$. Afterwards, M has to distinguish between finitely many possible candidate hypotheses for L, since $L = L_{\langle k,y+r\rangle}$ for some $r \in \mathbb{N}$ with $r \leq y$. By definition, each of these candidate hypotheses is uniquely characterized by infinitely many strings of form $a^k b^{r+1} a^m$. Thus, if M's actual guess is still incorrect then M eventually receives one of those strings and changes its mind to a correct guess for L. By construction, M repeats this guess in every subsequent stage. Hence, M learns L and Claim 1 is proved.

Claim 2. *No conservative IIM learns $\mathcal{C}$ with respect to $\mathcal{L}$.*

Adapting the proof technique developed in Lange and Zeugmann [15], Claim 2 follows by reducing the halting problem to $\mathcal{C} \in CONSV$ with respect to the hypothesis space $\mathcal{L}$. We omit the details. ∎

The latter result points to a particular strength of iterative IIMs. That is, iterative learning is not requested to realize the *subset principle* (cf. [24]). Moreover, the proof of the latter theorem shows that *redundancy* in the hypothesis space may lead to a serious increase in the learning power of iterative IIMs. Since $IT \subseteq FB$ as well as $IT \subseteq BEM_k$ for all $k \in \mathbb{N}$, the latter remarks apply to feed-back inference and bounded example memory identification, too. Consequently, one might be tempted to conjecture that even $IT \setminus CONSV \neq \emptyset$. This has also been claimed in Zeugmann and Lange [31, Theorem 19, Assertion (3)]. However, the proof given there is erroneous, and the stated conjecture is definitely false. Having the freedom to take a rich enough suitably chosen hypothesis space does really change the whole picture.

As it turned out, for proving $IT \subseteq CONSV$, $FB \subseteq CONSV$, and $BEM_k \subseteq CONSV$, it is conceptually simpler to use the characterization of conservative learning equating it with set-driven inference (cf. Lange and Zeugmann [16]). Set-drivenness describes the requirement that the output of an IIM is only allowed to depend on the range of its input.

Definition 6 (Wexler and Culicover [25]). *Let $\mathcal{C}$ be an indexable class.* ***An IIM M is said to be set-driven*** *with respect to $\mathcal{C}$ 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 positive presentations t, $\hat{t} \in \bigcup_{c \in \mathcal{C}} pos(c)$ provided $t_x^+ = \hat{t}_y^+$.*

Whenever the relevant indexable class $\mathcal{C}$ is clear from the context we refer to set-driven with respect to $\mathcal{C}$ as set-driven for short. By $s\text{-}LIM$ and $s\text{-}CONSV$ we denote the collection of all indexable classes that are LIM–inferable and $CONSV$–learnable, respectively, by some set-driven IIM. Moreover, whenever dealing with set-driven IIMs it is conceptionally advantageous to define or describe them in dependence on the relevant set obtained as input instead of the initial segments of a positive presentation usually fed an IIM.

The next proposition completely characterizes the learning capabilities of set-driven learners (cf. [16]).

Proposition 1. $s\text{-}LIM = CONSV = s\text{-}CONSV$.

Next, we show that every feed-back learner can be simulated by a set-driven IIM.

Theorem 2. $FB \subseteq s\text{-}LIM$.

Proof. Let $\mathcal{X}$ be the relevant learning domain over which $\mathcal{C}$ is defined, and assume $\mathcal{C} \in FB$. Then there are an IIM M and a hypothesis space $\mathcal{H} = (h_j)_{j \in \mathbb{N}}$ such that M witnesses the feed-back learnability of $\mathcal{C}$ with respect to $\mathcal{H}$. For proving $\mathcal{C} \in s\text{-}LIM$, first we construct a suitable hypothesis space $\hat{\mathcal{H}} = (\hat{h})_{j \in \mathbb{N}}$. Let $\mathcal{F} = F_0, F_1, F_2, \dots$ denote any repetition free enumeration of all finite subsets of $\mathcal{X}$. Furthermore, we assume an effective procedure computing for every finite set $F \subseteq \mathcal{X}$ its uniquely determined index $\#(F)$ in $\mathcal{F}$. We define

$$\hat{h}_j = \begin{cases} h_{j/2}, & \text{if} \quad j \text{ is even,} \\ F_{(j-1)/2}, & \text{if} \quad j \text{ is odd.} \end{cases}$$

Moreover, for every non-empty finite set $T \subseteq \mathcal{X}$ we define $rf(T) = x_0, x_1, \ldots, x_{\mathrm{card(T)}-1}$ to be the *repetition free* enumeration of all the elements of T in lexicographical order. By $\ell(T)$ we denote the lexicographically largest element of T. Finally, let $\sigma_0, \sigma_1, \sigma_2, \ldots$ be any effective enumeration of all finite sequences of elements from $\mathcal{X}$. The desired set-driven IIM $\hat{M}$ takes as its inputs finite sets T, and is defined as follows:

IIM $\hat{M}$: "On input T do the following:

Test for all $k \leq \mathrm{card}(T)$ whether or not $\sigma_k^+ \subseteq T$. For all k successively passing this test check whether or not the following condition is fulfilled for all $F \subseteq T$.

$M_{|\sigma_k|-1}(\sigma_k) = M_{|\sigma_k|+\mathrm{card}(F)-1}(\sigma_k \cdot rf(F)) = M_{|\sigma_k|+\mathrm{card}(F)}(\sigma_k \cdot rf(F) \cdot \ell(F))$.

If there exists a k passing this test, too, then choose the minimal one, compute $j = M_{|\sigma_k|-1}(\sigma_k)$, output $2j$, and request the next input.

Otherwise, output $2\#(T) + 1$, and request the next input."

By construction, $\hat{M}$ is set-driven and outputs in each learning step a hypothesis. It remains to show that $\hat{M}$ infers $\mathcal{C}$. Let $c \in \mathcal{C}$, and let $t \in pos(c)$. We distinguish the following cases.

Case 1. c is finite.

Then, there exists an $n \in \mathbb{N}$ such that $t_n^+ = c$. Thus, it suffices to show that $c = \hat{h}_{\hat{M}(c)}$. In case $\hat{M}$ outputs $2\#(c)+1$, we are done. Otherwise, $\hat{M}$ has found a finite sequence σ with $\sigma^+ \subseteq c$ that in particular fulfills $M_{|\sigma|-1}(\sigma) = M_{|\sigma|+m-1}(\sigma \cdot rf(c)) = M_{|\sigma|+m}(\sigma \cdot rf(c) \cdot \ell(c))$, where $m = \mathrm{card}(c)$. Hence, $\hat{M}$ computes $j = M_{|\sigma|-1}(\sigma)$ and outputs $2j$. Since $\hat{h}_{2j} = h_j$, it suffices to show $c = h_j$. Obviously, $\hat{t} = \sigma \cdot rf(c) \cdot \ell(c), \ell(c), \ldots$ constitutes a positive presentation for c. We know that $j = M_{|\sigma|+m-1}(\sigma \cdot rf(c)) = M_{|\sigma|+m}(\sigma \cdot rf(c) \cdot \ell(c))$. Now, let $Q(j, \ell(c))$ be the query computed by M in Stage $|\sigma| + m$. By construction, this query equals the query computed in Stage $|\sigma| + m + 1$. Since M has already seen all elements belonging to c, it must receive in both stages the same answer. But this implies that M computes in every subsequent stage the query $Q(j, \ell(c))$, too, thereby always receiving the same answer. Consequently, $(M_n(\hat{t}))_{n\in\mathbb{N}}$ converges to j. Since M *FB*–learns c, we have $c = h_j$.

Case 2. c is infinite.

Since M has to learn c from every positive presentation for it, there exists a sequence σ for c such that

$$M_{|\sigma|-1}(\sigma) = M_{|\sigma|+\mathrm{card}(F)-1}(\sigma \cdot rf(F)) = M_{|\sigma|+\mathrm{card}(F)}(\sigma \cdot rf(F) \cdot \ell(F)) \tag{1}$$

for all finite subsets $F \subseteq c$. In particular, every locking sequence must fulfil Condition (1) (cf. [18]). For proving $\hat{M}$'s correctness, we fix the sequence σ satisfying (1) that is first enumerated, i.e., let $\sigma = \sigma_{m_0}$, and for all σ_z, $z < m_0$, with $\sigma_z^+ \subseteq c$ we have σ_z does not fulfil (1).

Let t^c be the lexicographically ordered positive presentation for c. Then $t_n^c = rf(t_n^{c,+})$ for all $n \in \mathbb{N}$. Consequently, $j = M_{|\sigma|-1}(\sigma) = M_{|\sigma|+n}(\sigma \cdot t_n^c)$ for all $n \in \mathbb{N}$. Thus M

converges to j when fed $\sigma \cdot t^c$. Moreover, $h_j = c$, since M FB–infers c. Now, it suffices to show that $\hat{M}$ converges to $2j$. Let $t \in pos(c)$ be arbitrarily fixed, let n_0 be the least index n such that $\sigma^+ \subseteq t_n^+$, and let $m = \max\{n_0, m_0\}$. Hence, on every input t_{m+r}^+, $r \in \mathbb{N}$, $\hat{M}$ finds at least one k fulfilling the tests described. Moreover, since σ_{m_0} is the first enumerated sequence for c satisfying (1), for every other sequence σ_z, $z < m_0$, with $\sigma_z^+ \subseteq c$ there must be a finite set $F \subseteq c$ such that

$$M_{|\sigma_z|-1}(\sigma_z) = M_{|\sigma_z|+\text{card}(F)-1}(\sigma_z \cdot rf(F)) = M_{|\sigma_z|+\text{card}(F)}(\sigma_z \cdot rf(F) \cdot \ell(F))$$

is not fulfilled. Consequently, $\hat{M}$ converges to $2M_{|\sigma|-1}(\sigma) = 2j$, and we are done. ∎

The proof technique developed above is also powerful enough to establish the analogous result for bounded example memory inference. This is done in the next theorem.

THEOREM 3. $\bigcup_{k\in\mathbb{N}} BEM_k \subseteq CONSV$.

Proof. We use the same notations as in the proof of Theorem 2. In particular, let $\mathcal{C} \in \bigcup_{k\in\mathbb{N}} BEM_k$. Hence, there exists a $k \in \mathbb{N}$ such that $\mathcal{C} \in BEM_k$. Furthermore, let the hypothesis space $\hat{\mathcal{H}}$ be defined as above. The desired set-driven IIM $\hat{M}$ is essentially defined as in the proof of Theorem 2. However, we have to modify it appropriately to handle the information contained in the example memory.

IIM $\hat{M}$: "On input T do the following:
Test for all $y \leq \text{card}(T)$ whether or not $\sigma_y^+ \subseteq T$. For all y successively passing this test check whether or not the following condition is fulfilled for all $F \subseteq T$.

$$\begin{aligned}\pi_1(M_{|\sigma_y|-1}(\sigma_y)) &= \pi_1(M_{|\sigma_y|+\text{card}(F)-1}(\sigma_y \cdot rf(F))) \\ &= \pi_1(M_{|\sigma_y|+\text{card}(F)}(\sigma_y \cdot rf(F) \cdot \ell(F))) \\ &= \ldots = \pi_1(M_{|\sigma_y|+\text{card}(F)+2k}(\sigma_y \cdot rf(F) \cdot \underbrace{\ell(F) \cdot \ldots \cdot \ell(F)}_{(2k+1)\text{ times}}))\end{aligned}$$

If there exists a y passing this test, too, then choose the minimal one, compute $j = \pi_1(M_{|\sigma_y|-1}(\sigma_y))$, output $2j$, and request the next input.
Otherwise, output $2\#(T) + 1$, and request the next input."

By construction, $\hat{M}$ is set-driven and outputs in each learning step a hypothesis. It remains to show that $\hat{M}$ infers $\mathcal{C}$. Let $c \in \mathcal{C}$, and let $t \in pos(c)$. We distinguish the following cases.

Case 1. c is finite.

Then, there exists an $n \in \mathbb{N}$ such that $t_n^+ = c$. Thus, it suffices to show that $c = \hat{h}_{\hat{M}(c)}$. In case $\hat{M}$ outputs $2\#(c) + 1$, we are done. Otherwise, $\hat{M}$ has found a finite sequence σ with $\sigma^+ \subseteq c$ that in particular fulfills

$$\begin{aligned}\pi_1(M_{|\sigma|-1}(\sigma)) &= \pi_1(M_{|\sigma|+m-1}(\sigma \cdot rf(c))) = \pi_1(M_{|\sigma|+m}(\sigma \cdot rf(c) \cdot \ell(c))) \\ &= \ldots = \pi_1(M_{|\sigma|+m+2k}(\sigma \cdot rf(c) \cdot \ell(c)^{2k+1})),\end{aligned}$$

where $m = \mathrm{card}(c)$. Hence, $\hat{M}$ computes $j = \pi_1(M_{|\sigma|-1}(\sigma))$ and outputs $2j$. Since $\hat{h}_{2j} = h_j$, it suffices to show $c = h_j$. Obviously, $\hat{t} = \sigma \cdot rf(c) \cdot \ell(c), \ell(c), \ldots$ constitutes a positive presentation for c. We know that

$$\begin{aligned} j = \pi_1(M_{|\sigma|+m-1}(\sigma \cdot rf(c))) &= \pi_1(M_{|\sigma|+m}(\sigma \cdot rf(c) \cdot \ell(c))) \\ &= \ldots = \pi_1(M_{|\sigma|+m+2k}(\sigma \cdot rf(c) \cdot \ell(c)^{2k+1})). \end{aligned} \tag{2}$$

We claim that $j = \pi_1(M_{|\sigma|+m+2k+r}(\sigma \cdot rf(c) \cdot \ell(c)^{2k+r+1}))$ for all $r \in \mathbb{N}$. Having this claim, we know that M converges on $\hat{t}$, and we are done.

Clearly, M can change its hypothesis only in case it computes and outputs a set S not yet tested. Let $S_{|\sigma|+m-1}$, $S_{|\sigma|+m}$, $\ldots$, $S_{|\sigma|+m+2k}$ be the sets output in Stage $|\sigma| + m - 1$, $|\sigma| + m$, ..., $|\sigma| + m + 2k$, respectively. By definition of BEM_k we have $S_{|\sigma|+m+\mu} \subseteq S_{|\sigma|+m+\mu-1} \cup \{\ell(c)\}$ for all $\mu = 0, \ldots, 2k$. Consequently, if M once excludes an element $s \neq \ell(c)$ from the set received as actual input, it cannot include this element again. It may, however, include $\ell(c)$ and exclude it afterwards. Nevertheless, if there are two sets, say $S_{|\sigma|+m+r_1}$ and $S_{|\sigma|+m+r_2}$ such that $S_{|\sigma|+m+r_1} = S_{|\sigma|+m+r_2}$ and $r_1 \neq r_2$, then M will produce a periodic sequence of sets, and Equation 2 implies that the first components of all hypotheses output afterwards are equal, too. Now, it suffices to argue that this event must happen. Since $\mathrm{card}(S_{|\sigma|+m}) \leq k$, M can exclude at most k elements from the set $S_{|\sigma|+m}$ output in Stage $|\sigma| + m$. Additionally, whenever an element is excluded, it may include $\ell(c)$, and it may exclude $\ell(c)$. Hence, the longest sequence of pairwise non-equal sets has length $2k + 1$. Finally, since the last element of $rf(c)$ equals $\ell(c)$, the IIM $\hat{M}$ has tested a sequence of length $2k + 2$, and hence it must have found a period.

Case 2. c is infinite.

Since M has to BEM_k–infer c there exists a locking sequence σ for c (cf. [18]), and therefore

$$\begin{aligned} \pi_1(M_{|\sigma|-1}(\sigma)) &= \pi_1(M_{|\sigma|+\mathrm{card}(F)-1}(\sigma \cdot rf(F))) = \pi_1(M_{|\sigma|+\mathrm{card}(F)}(\sigma \cdot rf(F) \cdot \ell(F))) \\ &= \ldots = \pi_1(M_{|\sigma|+\mathrm{card}(F)+2k}(\sigma \cdot rf(F) \cdot \ell(F)^{2k+1})) \end{aligned} \tag{3}$$

for all finite subsets $F \subseteq c$. We fix the sequence σ satisfying (3) that is first enumerated, i.e., let $\sigma = \sigma_{m_0}$, and all σ_z, $z < m_0$, with $\sigma_z^+ \subseteq c$ do not fulfil Condition (3).

Let t^c be the lexicographically ordered positive presentation for c. Then $t_n^c = rf(t_n^{c,+})$ for all $n \in \mathbb{N}$. Consequently, $j = \pi_1(M_{|\sigma|-1}(\sigma)) = \pi_1(M_{|\sigma|+n}(\sigma \cdot t_n^c))$ for all $n \in \mathbb{N}$. Thus M converges to j when fed $\sigma \cdot t^c$, and $h_j = c$, since M BEM_k–infers c. Now, we show that $\hat{M}$ converges to $2j$. Let $t \in pos(c)$, let n_0 be the least index n such that $\sigma^+ \subseteq t_n^+$, and let $m = \max\{n_0, m_0\}$. Therefore, on every input t_{m+r}^+, $r \in \mathbb{N}$, $\hat{M}$ finds at least one y fulfilling the tests described. Taking the choice of σ_{m_0} into account, we conclude that for every other sequence σ_z, $z < m_0$, with $\sigma_z^+ \subseteq c$ there must be a finite set $F \subseteq c$ such that

$$\begin{aligned} \pi_1(M_{|\sigma_z|-1}(\sigma_z)) &= \pi_1(M_{|\sigma_z|+\mathrm{card}(F)-1}(\sigma_z \cdot rf(F))) = \pi_1(M_{|\sigma_z|+\mathrm{card}(F)}(\sigma_z \cdot rf(F) \cdot \ell(F))) \\ &= \ldots = \pi_1(M_{|\sigma_z|+\mathrm{card}(F)+2k}(\sigma_z \cdot rf(F) \cdot \ell(F)^{2k+1})) \end{aligned}$$

is not fulfilled. Consequently, $\hat{M}$ converges to $2M_{|\sigma|-1}(\sigma) = 2j$. ∎

The latter theorems immediately allow the corollary that every iterative IIM can be simulated by a set-driven IIM, too. Nevertheless, we present a separate proof for it, since the construction considerably simplifies. Note that Kinber and Stephan [8] also proved this result in the setting of learning recursively enumerable languages.

COROLLARY 4. $IT \subseteq s\text{-}LIM$.

Proof. We use the same notations as in the proof of Theorem 2. The desired set-driven IIM $\hat{M}$ is defined as follows:

IIM $\hat{M}$: "On input T do the following: Compute $rf(T)$, and $\ell(T)$. Check whether or not $M_{\mathrm{card(T)}-1}(rf(T)) = M_{\mathrm{card(T)}}(rf(T) \cdot \ell(T))$.
If it is, output $2M_{\mathrm{card(T)}-1}(rf(T))$, and request the next input.
Otherwise, output $2\#(T)+1$, and request the next input."

By definition, $\hat{M}$ is set-driven. For showing that $\hat{M}$ LIM–infers $\mathcal{C}$ with respect to $\hat{\mathcal{H}}$, let $c \in \mathcal{C}$, and let $t \in pos(c)$. We distinguish the following cases.

Case 1. c is finite.

If $\hat{M}(c) = 2\#(c)+1$ we are done by construction. Otherwise, $M_{\mathrm{card(c)}-1}(rf(c)) = M_{\mathrm{card(c)}}(rf(c) \cdot \ell(c))$. Let $j = M_{\mathrm{card(c)}-1}(rf(c))$; then rewriting $M_{\mathrm{card(c)}}(rf(c) \cdot \ell(c))$ yields $M_{\mathrm{card(c)}}(rf(c) \cdot \ell(c)) = M(j, \ell(c)) = M(j, \ell(c)) = M_{\mathrm{card(c)}+1}(rf(c) \cdot \ell(c) \cdot \ell(c))$. Hence, M converges on the positive presentation $rf(c) \cdot \ell(c), \ell(c), \ldots \in pos(c)$. Since M learns c, we are again done.

Case 2. c is infinite.

Let $t^c = w_0, w_1, w_2, \ldots$ be the lexicographically ordered positive presentation for c. Since M IT–learns c from t^c, there exists an $n_0 \in \mathbb{N}$ such that $M_{n_0}(t^c) = M_n(t^c)$ for all $n \geq n_0$. Moreover, $M_{n_0+1}(t^c_{n_0} \cdot w) = M_{n_0}(t^c)$ for all $w \in c \setminus t^{c,+}_{n_0}$. For seeing this, let $n_0 + z$ be the uniquely determined index of w in t^c, i.e., $w = w_{n_0+z}$. Letting $j = M_{n_0}(t^c)$ we have

$$j = M_{n_0+1}(t^c) = M(j, w_{n_0+1}) = \ldots = M(j, w_{n_0+z}) = M_{n_0+z}(t^c).$$

Consequently, there is an initial segment $\sigma = t^c_{n_0}$ of $t^c \in pos(c)$ on which M is locked provided this segment is extended with elements $w \in c \setminus t^{c,+}_{n_0}$. Since M learns c from t^c we have $c = h_j = \hat{h}_{2j}$. Finally, since $t \in pos(c)$ there exists an index m_0 such that $\sigma^+ \subseteq t^+_{m_0}$. Thus, σ is a prefix of $rf(t^+_{m_0})$, and hence $\hat{M}(t^+_m) = 2j$ for all $m > m_0$. ∎

We finish this subsection by proving the following upper bound for the learning capabilities of iterative learning, feed-back inference, and bounded example memory identification.

THEOREM 5.

(1) $IT \subset CONSV$,

(2) $FB \subset CONSV$,

(3) $\bigcup_{k \in \mathbb{N}} BEM_k \subset CONSV$.

Proof. Since $IT \subseteq FB$ as well $IT \subseteq BEM_k$ for every $k \in \mathbb{N}$, Assertion (1) directly follows from (2) or (3). Furthermore, by Theorems 2 and 3 we obtain from Proposition 1 the containment of FB and $\bigcup_{k\in\mathbb{N}} BEM_k$ in $CONSV$, respectively. Thus, it suffices to prove that the stated inclusions are proper.

Claim 1. $CONSV \setminus FB \neq \emptyset$.

Let $L = \{a\}^+$, and for all $j,k \in \mathbb{N}^+$ let $L_{j,k} = \{a^m \mid 1 \le m \le j\} \cup \{b^j, a^k\}$. Let $\mathcal{C}_{FB}$ and $\mathcal{H}$ be the collection and canonical enumeration of all the languages $L_{j,k}$ and L, respectively. One straightforwardly verifies $\mathcal{C}_{FB} \in CONSV$ with respect to $\mathcal{H}$.

Next, we show $\mathcal{C}_{FB} \notin FB$. Suppose there is an IIM M which FB–learns $\mathcal{C}_{FB}$. Let $\sigma = s_0, \dots, s_n$ be a locking sequence for $L = \{a\}^+$ (cf. [18]). Furthermore, let $z = \max\{m \mid a^m \in \sigma^+\}$, and let $j_n = M_n(\sigma)$. Thus, when fed any finite extension of σ with strings from L, M repeats its guess j_n.

Now we select two different finite languages $\hat{L}$ and $\tilde{L}$ from $\mathcal{C}_{FB}$ and show that M fails to learn at least one of them. Set $\hat{L} = \{a^m \mid 1 \le m \le z\} \cup \{b^z\}$, and consider M's behavior when successively fed the following text $\hat{t} \in pos(\hat{L})$. Let $\hat{t} = \sigma \cdot a, \dots, a^z, b^z, a, a, \dots$ Since M learns $\hat{L}$ from $\hat{t}$, there is an $r > n + z + 1$ such that M, after processing the initial segment $\hat{t}_r$, does not perform any further mind change. By definition, $\hat{t}_{y+1} = \hat{t}_y \cdot a$ for all $y \ge r$. Consequently, after M has processed the initial segment $\hat{t}_r$ it must always make the same query. Thus, M makes at most finitely many different queries when fed $\hat{t}$. Now, choose any string $\tilde{s} \in L \setminus \hat{L}$ the IIM M is never asking for, and set $\tilde{L} = \hat{L} \cup \{\tilde{s}\}$.

Finally, let $\tilde{t} = \sigma \cdot \tilde{s}, a, \dots, a^z, b^z, a, a, \dots$, and thus $\tilde{t} \in pos(\tilde{L})$. Since σ is a locking sequence for L, M outputs j_n after processing the initial segment $\sigma \cdot \tilde{s}, a, \dots, a^z$. Afterwards, M receives the string b^z. Based on the local input j_n and b^z, M must make the same query, say s', as in case the initial segment $\sigma \cdot a, \dots, a^z, b^z$ has been processed. Since $s' \neq \tilde{s}$, M obtains the same answer and generates, therefore, the same guess as in the former case. Now, the same argument may be iterated in order to show that M, when successively fed $\tilde{t}$, generates afterwards the same sequence of hypotheses as in case it is processing $\hat{t}$. Hence, M is fooled.

Claim 2. $CONSV \setminus \bigcup_{k\in\mathbb{N}} BEM_k \neq \emptyset$.

For all $j \in \mathbb{N}$, let $L_j = \{a\}^+ \setminus \{a^{j+1}\}$. Let $\mathcal{C}_{CONSV}$ and $\mathcal{H} = (L_j)_{j\in\mathbb{N}}$ be the collection and canonical enumeration of all these languages L_j, respectively. Again, $\mathcal{C}_{CONSV} \in CONSV$ can be easily verified.

It remains to show that $\mathcal{C}_{CONSV} \notin \bigcup_{k\in\mathbb{N}} BEM_k$. Suppose the converse, i.e., there are a $k \in \mathbb{N}$, an IIM M, and a hypothesis space $\mathcal{H} = (h_j)_{j\in\mathbb{N}}$ such that M BEM_k–infers $\mathcal{C}_{CONSV}$ with respect to $\mathcal{H}$. Recall that M has to compute its actual guess j_n from its previously made one, say j_{n-1}, a set $S_{n-1} \subseteq t^+_{n-1}$ of cardinality k, and the next string s_n. The basic idea behind the proof is easily explained. The set S_{n-1} does not contain sufficient information in the following case: M first reads a locking sequence σ for L_0 and then further strings belonging to L_0. Note that, after having read σ, M cannot encode any additional information in its actual guesses until the string $a \notin L_0$ possibly appears in the positive presentation. Thus, it has to memorize all *relevant* data in the actual output sets S. But their cardinality is bounded by k. Thus, there are at least two different extensions of σ with elements from L_0 on which M outputs the same hypothesis $\langle j, S\rangle$. Consequently, M can be forced to forget some *relevant* information which makes it impossible for it to infer some language $L_j \neq L_0$.

We continue with the formal proof. Since M has to infer $L_0 = \{a\}^+ \setminus \{a\}$, there has to be a locking sequence σ for L_0 (cf. [18]). Hence, we know that $\pi_1(M_{|\sigma|-1}(\sigma)) = \pi_1(M_{|\sigma|+r}(\sigma \cdot t))$ for all $t \in pos(L_0)$ and all $r \in \mathbb{N}$. For every $\emptyset \neq D \subseteq \{a\}^+$, let $rf(D) = s_0, \ldots, s_{\mathrm{card(D)}-1}$ be the repetition free enumeration of D in lexicographical order. Taking into account that there are more than 2^n sets $D \subseteq \{a^2, a^3, \ldots, a^{|\sigma|+3n}\}$ of cardinality $|\sigma| + 2n$ but only $O(n^k)$ many pairwise different subsets of cardinality at most k there must be sets D_1 and D_2 such that $M_{2|\sigma|+2n-1}(\sigma \cdot rf(D_1)) = M_{2|\sigma|+2n-1}(\sigma \cdot rf(D_2))$, and $D_1 \neq D_2$ provided n is sufficiently large.

Now we are ready for showing that M fails to identify $\mathcal{C}_{CONSV}$. We define two languages $\hat{L},\ \tilde{L} \in \mathcal{C}_{CONSV}$ witnessing M's weakness. Let $\tau = rf(D_1)$, and let $\rho = rf(D_2)$. By construction $|\tau| = |\rho|$ but $\tau^+ \neq \rho^+$. Hence, there are $\hat{w} \in \rho^+ \setminus \tau^+$ and any $\tilde{w} \in \tau^+ \setminus \rho^+$. Set $\hat{L} = L_0 \setminus \{\hat{w}\}$ and $\tilde{L} = L_0 \setminus \{\tilde{w}\}$. Let t' be any positive presentation of $L_0 \setminus \{\hat{w}, \tilde{w}\}$. Obviously, $\hat{t} = \sigma \cdot \tau \cdot t'$ defines a positive presentation of $\hat{L}$, and $\tilde{t} = \sigma \cdot \rho \cdot t'$ belongs to $pos(\tilde{L})$. As shown above $M_{2|\sigma|+2n-1}(\hat{t}) = M_{2|\sigma|+2n-1}(\tilde{t})$. By construction, past point $2|\sigma| + 2n - 1$, both texts are identical. By definition of BEM_k, we may therefore conclude that $M_{2|\sigma|+2n+z-1}(\hat{t}) = M_{2|\sigma|+2n+z-1}(\tilde{t})$ for all $z \geq 0$, too. Consequently, M fails to learn $\hat{L}$ or $\tilde{L}$ when fed $\hat{t}$ and $\tilde{t}$, respectively, a contradiction. Thus Claim 2 is proved. ∎

3.2. *On the Strength of Incremental Learning*

Now we study to what extend, if ever, feed-back learning and bounded example memory inference, respectively, enlarges the learning capabilities of iterative learning. Fulk *et al.* [7] proved that even the ability to store exactly one distinguished example seriously increases the learning capabilities of iterative IIMs. Furthermore, they stated without proof that $BEM_k \subset BEM_{k+1}$ for all $k \in \mathbb{N}$. We provide a proof for the latter statement. Moreover, allowing the learner to ask whether or not a particular element did already appear in the initial segment of the positive presentation processed so far results in a remarkable increase of learning power, too. However, bounded example memory inference and feed-back learning extent the learning capabilities of iterative identification in different directions as we shall see. Within the following theorem we summarize the results obtained by Fulk *et al.* [7] as well as the superiority of feed-back learning over iterative inference.

THEOREM 6.

(1) $IT \subset FB$,

(2) $IT \subset BEM_1$,

(3) $BEM_k \subset BEM_{k+1}$ *for all* $k \in \mathbb{N}$.

Proof. Since $IT \subseteq FB$, $IT = BEM_0$, and $BEM_k \subseteq BEM_{k+1}$ for all $k \in \mathbb{N}$, it suffices to show that all the stated inclusions are proper.

CLAIM 1. $FB \setminus IT \neq \emptyset$.

The indexable class $\mathcal{C}_{CONSV}$ used in the proof of Theorem 5 witnesses the desired separation, too. Since $\mathcal{C}_{CONSV} \notin \bigcup_{k\in\mathbb{N}} BEM_k$, it suffices to prove $\mathcal{C}_{CONSV} \in FB$. Let $\mathcal{H} = (L_j)_{j\in\mathbb{N}}$, where $L_j = \{a\}^+ \setminus \{a^{j+1}\}$. Let $L \in \mathcal{C}_{CONSV}$, and let $t = (s_j)_{j\in\mathbb{N}} \in pos(L)$. The desired feed-back IIM M is defined in stages, where Stage n conceptually describes M_n.

Stage 0. On input s_0 do the following.
Set $j_0 = 0$, output j_0, and go to Stage 1.

Stage n, $n \geq 1$. On input j_{n-1} and s_n do the following.
Make the query '$a^{1+j_{n-1}}$.' If the answer is 'NO,' set $j_n = j_{n-1}$, output j_n, and go to Stage $n+1$.
If the answer is 'YES,' set $j_n = j_{n-1}+1$, output j_n, and go to Stage $n+1$.

Let $L = L_z = \{a\}^+ \setminus \{a^{z+1}\}$. Since t is a text for L, there is a least $n \in \mathbb{N}$ such that the initial segment t_n contains all strings a^m with $1 \leq m \leq z$. By definition, M asks the oracle in Stage $n+z$ whether or not the string a^{z+1} has already been presented. Since $a^{z+1} \notin L$, the reply is 'NO'. Thus, M repeats this query in every subsequent stage. Finally, the oracle's reply enables M to output the correct guess z. Hence, M FB–learns L from t.

CLAIM 2. $BEM_1 \setminus IT \neq \emptyset$.

The proof idea used here is similar to that one developed in Fulk *et al.* [7] who proved the same result. We include this proof here, since the general proof of Assertion (3) is based on it. Recall the definition of the indexable class $\mathcal{C}_{FB}$ introduced in the demonstration of Theorem 5, Claim 1. That is, $\mathcal{C}_{FB}$ is the collection consisting of the language $L = \{a\}^+$ and of all the languages $L_{j,k} = \{a^m \mid 1 \leq m \leq j\} \cup \{b^j, a^k\}$ for $j,k \in \mathbb{N}^+$. We set $\mathcal{C}_{BEM_1} = \mathcal{C}_{FB}$, and show that $\mathcal{C}_{BEM_1}$ witnesses the desired separation. Since $\mathcal{C}_{BEM_1} \notin FB$ (cf. the demonstration of Theorem 5, Claim 1) and $IT \subseteq FB$, it suffices to show that $\mathcal{C}_{BEM_1} \in BEM_1$.

Let $\mathcal{H}_{BEM_1} = (h_j)_{j\in\mathbb{N}}$ be the canonical enumeration of all the languages in $\mathcal{C}_{BEM_1}$. We define an IIM M which BEM_1–identifies $\mathcal{C}_{BEM_1}$ with respect to $\mathcal{H}_{BEM_1}$. Let $L' \in \mathcal{C}_{BEM_1}$, and let $t = (s_j)_{j\in\mathbb{N}} \in pos(L')$. The IIM M is defined in stages, where Stage n conceptually describes M_n.

Stage 0. On input s_0 do the following.
If $s_0 = a^\ell$ for some $\ell \in \mathbb{N}^+$, then set $S_0 = \{a^\ell\}$ and compute the canonical index z for $L = \{a\}^+$.
Otherwise, i.e., $s_0 = b^\ell$ for some $\ell \in \mathbb{N}^+$, set $S_0 = \emptyset$, and determine the canonical index z for $L_{\ell,1} = \{a^m \mid 1 \leq m \leq \ell\} \cup \{b^\ell\}$.
Set $j_0 = z$, output $\langle j_0, S_0\rangle$, and go to Stage 1.

Stage n, $n \geq 1$. On input $\langle j_{n-1}, S_{n-1}\rangle$ and s_n do the following.
If $s_n = b^\ell$ for some $\ell \in \mathbb{N}^+$, then determine the canonical index z for the language $L_{\ell,1} \cup S_{n-1}$. Set $j_n = z$, $S_n = \emptyset$, output $\langle j_n, S_n\rangle$, and go to Stage $n+1$.

If $s_n = a^\ell$ for some $\ell \in \mathbb{N}^+$, test whether $S_{n-1} = \emptyset$. If it is, then determine the canonical index z for the language $h_{j_{n-1}} \cup \{a^\ell\}$, and set $S_n = \emptyset$ and $j_n = z$.
If $S_{n-1} \neq \emptyset$, then set $j_n = j_{n-1}$. Let $S_{n-1} = \{s\}$; check whether $|s_n| \geq |s|$. If it is,

set $S_n = \{s_n\}$. Otherwise, let $S_n = \{s\}$.
Output $\langle j_n, S_n\rangle$, and go to Stage $n+1$.

It remains to show that M infers L'. Obviously, if $L' = L$, then M outputs in every stage a correct guess for L'. Now, let $L' = L_{j,k}$ for some $j, k \in \mathbb{N}^+$. Hence, there must be an n such that $s_n = b^j$. By construction, S_{n-1} contains the longest string of form a^m, say s, which has been presented so far. If $|s| > |s_n|$ or $L' = L_{j,1}$, then M outputs in this and every subsequent stage a correct guess for L'. Otherwise, M guesses the finite sublanguage $L_{j,1}$ of L' in Stage n. Since $t \in pos(L')$, M eventually receives in some subsequent stage the one and only missing string a^k, $k > j$. Then it changes its mind to a correct guess for $L' = L_{j,1} \cup \{a^k\}$ that is repeated in every subsequent stage. Thus, M BEM_1–learns L', and Claim 2 is shown.

Claim 3. *$BEM_{k+1} \setminus BEM_k \neq \emptyset$ for all $k \in \mathbb{N}$.*

Because of $BEM_0 = IT$, the $k = 0$ case has already been shown (cf. Claim 2). The general case is handled by enlarging the indexable class $\mathcal{C}_{BEM_1}$. Let $k \in \mathbb{N}^+$, and let $\langle \ell_0, \ldots, \ell_k\rangle$ be any $k+1$-tuple of elements from $\mathbb{N}^+$. For every $j \in \mathbb{N}^+$ let $L_{\langle j,\ell_0,\ldots,\ell_k\rangle} = \{a^m \mid 1 \leq m \leq j\} \cup \{b^j, a^{\ell_0}, \ldots, a^{\ell_k}\}$, and let $L = \{a\}^+$. By $\mathcal{C}_{BEM_{k+1}}$ and $\mathcal{H}_{BEM_{k+1}}$ we denote the collection and canonical enumeration of all the languages $L_{\langle j,\ell_0,\ldots,\ell_k\rangle}$ and L, respectively. By definition, $\mathcal{C}_{BEM_1} \subseteq \mathcal{C}_{BEM_{k+1}}$.

$\mathcal{C}_{BEM_{k+1}} \in BEM_{k+1}$ with respect to $\mathcal{H}_{BEM_{k+1}}$ can be shown using a minor modification of the IIM M defined in Claim 2 above. As long as no string of the form b^m occurs, the modified IIM $\hat{M}$ simply stores the $(k+1)$th longest strings of the form a^n seen so far, and outputs the canonical index for $\{a\}^+$ along with this set. If a string s of the form b^m appears, M outputs the canonical index for the least language $\hat{L}$ which contains both s and the $(k+1)$th longest string from $\{a\}^+$ seen so far. Past that point, there is no need to store any further string, since the target language has to be a finite superset of $\hat{L}$, and moreover, in case that $\hat{L}$ does not equal the target language, the missing strings have to appear in some subsequent step. We omit further details.

Now we prove the remaining part, i.e., $\mathcal{C}_{BEM_{k+1}} \notin BEM_k$. Suppose the converse, i.e., there are a hypothesis space $\mathcal{H}$ and a BEM_k IIM M inferring $\mathcal{C}_{BEM_{k+1}}$ with respect to $\mathcal{H}$. The basic idea is similar to that one used in the demonstration of Theorem 5, Claim 2. Since M learns $L = \{a\}^+$, there has to be a locking sequence σ for L (cf. [18]). Hence, we know that, $j = \pi_1(M_{|\sigma|-1}(\sigma)) = \pi_1(M_{|\sigma|+r}(\sigma \cdot t))$ for all $t \in pos(L)$ and all $r \in \mathbb{N}$.

Let $d = \max\{|s| \mid s \in \sigma^+\}$, let $n > k$, and consider all sets

$$D = \sigma^+ \cup \{a^{n_1}, a^{n_2} \ldots, a^{n_{k+1}}\} \text{ with } d < n_1 < n_2 < \ldots < n_{k+1} \leq d + 3n. \tag{4}$$

There are $\binom{3n}{k+1}$ such sets D. On the other hand, there exist only $\sum_{j=0}^{k} \binom{3n}{j}$ many pairwise different subsets $S \subseteq \{a^{n_1}, a^{n_2} \ldots, a^{n_{k+1}}\}$ with $d < n_1 < n_2 < \ldots n_{k+1} \leq d + 3n$ of cardinality at most k. Taking into account that $3n - m \geq 2n$ for all $m \leq k$ and using the well-known estimate

$$\sum_{j=0}^{k} \binom{3n}{j} \leq \left(\frac{3en}{k}\right)^k \text{ for all } 3n \geq k \geq 1,$$

we obtain

$$\frac{\binom{3n}{k+1}}{\sum\limits_{j=0}^{k}\binom{3n}{j}} > \frac{2^{k+1}nk^k}{(k+1)!(3e)^k} > 1 \text{ provided n is sufficiently large.} \tag{5}$$

Now, let again $rf(D)$ be the repetition free enumeration of D. Then (5) implies that there must be sets D_1, D_2 of type (4) such that $M_{|\sigma|+k}(\sigma \cdot rf(D_1)) = M_{|\sigma|+k}(\sigma \cdot rf(D_2))$ and $D_1 \neq D_2$, again provided n is sufficiently large.

Now we are ready for showing that M is not able to identify $\mathcal{C}_{BEM_{k+1}}$. Note that $|D_1| = |D_2| = k+1$. Since $D_1 \neq D_2$, there are strings $\hat{w} \in D_1 \setminus D_2$ and $\tilde{w} \in D_2 \setminus D_1$. We set $\hat{L} = \{a^m \mid 1 \leq m \leq d\} \cup D_1 \cup \{b^d\}$ and $\tilde{L} = \{a^m \mid 1 \leq m \leq d\} \cup D_2 \cup \{b^d\}$. Obviously, $\hat{L}, \tilde{L} \in \mathcal{C}_{BEM_{k+1}}$, and $\hat{L} \neq \tilde{L}$. Let t be any text for $\{a^m \mid 1 \leq m \leq d\}$, and consider $\hat{t} = \sigma \cdot rf(D_1) \cdot b^d \cdot t$ as well as $\tilde{t} = \sigma \cdot rf(D_2) \cdot b^d \cdot t$. By construction, $\hat{t} \in pos(\hat{L})$ and $\tilde{t} \in pos(\tilde{L})$ but $M_{|\sigma|+k}(\sigma \cdot rf(D_1)) = M_{|\sigma|+k}(\sigma \cdot rf(D_2))$. Past point $|\sigma|+k+1$ both text are identical, and thus M, if ever, converges on both $\hat{t}$ and $\tilde{t}$ to the same index. Hence, it must fail to learn at least one of the languages $\hat{L}$ and $\tilde{L}$. This proves Claim 3. ∎

The demonstration of Claim 2 above and of Claim 1 in Theorem 5 immediately imply that feed-back IIMs are not always able to compensate the additional learning capabilities of IIMs that are allowed to store exactly one example.

COROLLARY 7. $BEM_1 \setminus FB \neq \emptyset$.

However, the increase in learning power obtained by bounded examples memories and feed-back queries is *incomparable* as we shall see. Consequently, there is no unique way to design superior learning algorithms when space limitations are a serious concern. However, our overall goal is a bit more ambitious. We aim to compare the learning power of finite inference from *positive and negative data* (*FIN-INF*; cf. Definition 7 below) with those of bounded example memory learning and feed-back identification. As it turns out, feed-back learning from positive data can simulate finite inference from *positive and negative data* while bounded example memory learning cannot. This is interesting, since it addresses the issue whether information presentation can be traded versus memory limitations.

Next, we provide the formal definitions needed. Let $\mathcal{X}$ be a learning domain, and let $\mathcal{C}$ be any concept class defined over $\mathcal{X}$. Furthermore, let $c \in \mathcal{C}$, and let $i = (x_0, b_0), (x_1, b_1), \ldots$ be an infinite sequence of elements of $\mathcal{X} \times \{+, -\}$ such that range$(i) = \{x_k \mid k \in \mathbb{N}\} = \mathcal{X}$, $i^+ = \{x_k \mid (x_k, b_k) = (x_k, +), k \in \mathbb{N}\} = c$ and $i^- = \{x_k \mid (x_k, b_k) = (x_k, -), k \in \mathbb{N}\} = co\text{-}c = \mathcal{X} \setminus c$. Then we refer to i as an ***informant***. If c is classified via an informant then we also say that c is represented by ***positive and negative data***. Let c be a concept; by $info(c)$ we denote the set of all informants for c.

DEFINITION 7 (Gold, [11]). *Let $\mathcal{C}$ be an indexable class, let c be a concept, and let $\mathcal{H} = (h_j)_{j \in \mathbb{N}}$ be a hypothesis space.* ***An IIM M FIN-INF–identifies c from informant with respect to $\mathcal{H}$*** *iff for every informant i for c, there exists a $j \in \mathbb{N}$ such that M, when successively fed i outputs the single hypothesis j, $c = h_j$, and stops thereafter.*

Furthermore, M FIN-INF–identifies $\mathcal{C}$ with respect to $\mathcal{H}$ iff, for each $c \in \mathcal{C}$, M FIN-INF–identifies c with respect to $\mathcal{H}$.

The resulting learning type is denoted by FIN-INF.

We start our investigation of whether or not information presentation can be traded versus the mode of convergence and memory limitation with the following observation.

COROLLARY 8.

(1) *IT # FIN-INF*,

(2) $\bigcup_{k\in\mathbb{N}} BEM_k$ # *FIN-INF*.

Proof. Let $\mathcal{C}_{fin}$ be the class of all finite non-empty languages over the alphabet $\{a\}$. Obviously, $\mathcal{C}_{fin}$ witnesses $IT \setminus FIN\text{-}INF \neq \emptyset$.

For showing $FIN\text{-}INF \setminus \bigcup_{k\in\mathbb{N}} BEM_k \neq \emptyset$ we use the indexable class $\mathcal{C}_{CONSV}$. Since $\mathcal{C}_{CONSV} \notin \bigcup_{k\in\mathbb{N}} BEM_k$ (cf. Theorem 5, Claim 2), it remains to show that $\mathcal{C}_{CONSV} \in FIN\text{-}INF$. For that purpose recall that every language $L \in \mathcal{C}_{CONSV}$ is characterized by its uniquely determined negative example $x \in \{a\}^+ \setminus L$. Clearly, a finite learner has simply to wait until $(x, -)$ appears in the data. Then it outputs the canonical index of the corresponding language $L = \{a\}^+ \setminus \{x\}$ from $\mathcal{C}_{CONSV}$, and stops. ▮

The latter result is nicely contrasted by our next theorem establishing that feed-back learners capture the whole learning power of finite inference from positive and negative examples.

THEOREM 9. *FIN-INF* $\subset$ *FB*.

Proof. By definition $IT \subseteq FB$. Thus, $FB \setminus FIN\text{-}INF \neq \emptyset$ follows from Corollary 8. Next, we show $FIN\text{-}INF \subseteq FB$. Let $\mathcal{X}$ be the underlying learning domain, let $\mathcal{C} \in FIN\text{-}INF$, and let M be any IIM finitely learning $\mathcal{C}$ from informant with respect to some hypothesis space $\mathcal{H} = (h_j)_{j\in\mathbb{N}}$. Without loss of generality, we may assume that range$(\mathcal{H}) = \mathcal{C}$ (cf. Lange and Zeugmann [15]). Furthermore, we may assume M to be total, i.e., for every finite sequence of elements from $\mathcal{X} \times \{+, -\}$ M either outputs a hypothesis or it request the next input (cf. Lange *et al.* [17]).

The desired simulation is based on the following idea. The feed-back learner aims to find an initial segment of an informant for the target concept. Such an initial segment can be successively fed to the finitely learning IIM M until M stops or the segment is finished. If M makes an output (referred to as *ordinary hypothesis*), the feed-back learner maintains this guess as long as it is compatible with all the data provided afterwards. However, if an element x is presented that does not belong to the guessed concept then the whole process must be iterated. Since a feed-back learner is restricted to one query per one learning stage, we need *auxiliary hypotheses* to memorize the results of the queries made until a new possible initial segment of an informant for the target concept is found. Clearly, this idea will only work, if the strategy described above realizes the subset principle.

In order to design the desired feed-back IIM $\hat{M}$ we use the following hypothesis space $\hat{\mathcal{H}} = (\hat{h}_j)_{j\in\mathbb{N}}$. Let $\hat{h}_{2j} = h_j$ for all $j \in \mathbb{N}$, i.e., even indices describe the possible *ordinary hypotheses*. Odd indices are used for *auxiliary hypotheses*. For defining them, let $F_0,\ F_1,\ F_2,\ \dots$ be any effective enumeration of all finite subsets of $\mathcal{X}$ including the empty set. For for all $k, y \in \mathbb{N}$, set $\hat{h}_{2\langle k,y\rangle+1} = F_k \cup F_y$. The underlying semantics is as follows. The pair (F_k, F_y) represents the fact that all elements belonging to F_k have already been presented, whereas no element in F_y did appear in the data read so far. For the sake of readability, we use the following convention. When $\hat{M}$ is forced to output an auxiliary hypothesis, say $2\langle k, y\rangle + 1$, we use instead the phrase that $\hat{M}$ is outputting the pair (F_k, F_y). Let $x_0, x_1, x_2, \dots$ denote any effective repetition free enumeration of all and only the elements in $\mathcal{X}$. Given any pair (F_k, F_y) that satisfies $F_k \cap F_y = \emptyset$ and $F_k \cup F_y = \{x_j \mid j \leq z = \text{card}(F_k \cup F_y) - 1\}$; we set $i(F_k, F_y)_z = (x_0, b_0), \dots, (x_z, b_z)$, where, for all $j \leq z$, $b_j = 1$, if $x_j \in F_k$, and $b_j = 0$ in case that $x_j \in F_y$.

Next, we define the feed-back learner $\hat{M}$. Let $c \in \mathcal{C}$, and let $t = (s_n)_{n\in\mathbb{N}} \in pos(c)$. As usual, we define $\hat{M}$ in stages, where Stage n conceptually describes $\hat{M}_n$.

Stage 0. On input s_0 do the following.

Make the query 'x_0.' If the answer is 'NO,' then output the pair $(\emptyset, \{x_0\})$, and go to Stage 1. If the answer is 'YES,' then output the pair $(\{x_0\}, \emptyset)$, and go to Stage 1.

Stage n, $n \geq 1$. $\hat{M}$ receives as input j_{n-1} and the $(n+1)$th element s_n of t.

Case A. j_{n-1} is an ordinary hypothesis.

If $s_n \in \hat{h}_{j_{n-1}}$, set $j_n = j_{n-1}$, output j_n, and go to Stage $n+1$. Otherwise, make the query 'x_0.' If the answer is 'NO,' then output the pair $(\emptyset, \{x_0\})$, and go to Stage $n+1$. If the answer is 'YES,' then output the pair $(\{x_0\}, \emptyset)$, and go to Stage $n+1$.

Case B. j_{n-1} is an auxiliary hypothesis, say (P, N).

Test whether or not $s_n \in P \cup N$. If not, then set $P' = P$ and $N' = N$. Otherwise, set $P' = P \cup \{s_n\}$ and $N' = N \setminus \{s_n\}$. Determine $z = \text{card}(P' \cup N')$, and make the query '$x_z$.' If the answer is 'NO,' set $N' = N \cup \{x_z\}$. Else, set $P' = P \cup \{x_z\}$. Determine $i(P', N')_z$, and execute Instruction (β).

(β) Compute successively $M(i(P', N')_0)$, $M(i(P', N')_1)$, ..., $M(i(P', N')_z)$ until M outputs a hypothesis j, say on $i(P', N')_r$, $r \leq z$, or the whole initial segment has been fed. If there was no output or j does not fulfill $P' \subseteq \hat{h}_{2j}$ and $i(P', N')_r^- \cap \hat{h}_{2j} = \emptyset$, output the pair (P', N'), and go to Stage $n+1$. Otherwise, output the ordinary hypothesis $2j$ and go to Stage $n+1$.

Obviously, $\hat{M}$ is a feed-back IIM. We have to show that $\hat{M}$ learns the target concept c. First we make some helpful observations. Suppose $\hat{M}$ outputs in Stage n an auxiliary hypothesis, say (P, N). Let $z = \text{card}(P \cup N)$, and let P' and N' be the corresponding finite sets $\hat{M}$ has generated before executing Instruction (β) within Stage $n+1$. (* Note that $\text{card}(P' \cup N') = z + 1$. *) By construction, in all the Stages $n - z + 1, \dots, n - 1$ the IIM $\hat{M}$ has output auxiliary hypotheses, too. Hence, we have:

Observation 1. $P' \subseteq L$

Observation 2. $P' \cup N' = \{x_j \mid j \leq z\}$

Observation 3. For all $j \leq z$, $x_j \in t_n^+$ implies $x_j \in P'$.

For verifying the latter observation recall that $\hat{M}$ has successively queried $x_0, \dots, x_z$. Clearly, if x_ℓ, $1 \le \ell \le z$, has been presented before the query is made, then the answer is 'YES,' and thus $x_\ell \in P'$. Now assume x_ℓ is queried, say in Stage κ, $\kappa \le n$, but $x_\ell \notin t^+_\kappa$. Since $x_\ell \in t^+_{n+1}$, the element x_ℓ must appear as input in one of the Stages $\kappa+1, \dots, n+1$. However, then we are in Case B. Hence, x_ℓ has to be in P' and cannot belong to N'.

Furthermore, since two successively output auxiliary hypotheses are definitely different, $\hat{M}$ cannot converge to an auxiliary hypothesis.

Claim A. *$\hat{M}$ converges.*

Let $i(c)$ be the lexicographically ordered informant of c. Let y_0 be the least number y such that M after having successively fed $i(c)_y$ outputs a hypothesis, say j and stops. Since M has to learn c from $i(c)$ such a y must exist, and furthermore, $c = h_j$. Now, let n_0 be the least number n satisfying $i(c)^+_{y_0} \subseteq t^+_n$.

Suppose $\hat{M}$ has not yet converged when entering Stage n_0. Then there are two possible cases.

Case 1. j_{n_0-1} is an ordinary hypothesis.

Since $\hat{M}$ has not yet converged, this hypothesis must be changed in some subsequent stage, say in Stage $n_0 + \mu$, $\mu \ge 0$. Consequently, $\hat{M}$ begins in Stage $n_0 + \mu$ to query an initial segment of the lexicographically ordered informant for c by asking 'x_0,' and it outputs the corresponding auxiliary hypothesis (P, N). Note that the label of x_0 must be correct by the choice of n_0. In Stage $n_0 + \mu + 1$ the next query, i.e., 'x_1,' is made. Again, the resulting label of x_1 must be correct. Now, $\hat{M}$ enters Instruction (β). If $y_0 \le 1$, then M outputs j, and thus, $\hat{M}$ outputs $2j$. Since $c = \hat{h}_{2j}$, this guess is repeated in every subsequent stage. Consequently, $\hat{M}$ converges. On the other hand, if $y_0 > 1$, then, by the choice of y_0, M cannot make an output. Thus, $\hat{M}$ outputs the auxiliary hypothesis correctly describing the labeling of x_0 and x_1. Iterating this argument, we see that $\hat{M}$ outputs exclusively auxiliary hypotheses until the correct initial segment $i(c)_{y_0}$ is obtained. Then, entering Instruction (β) results in obtaining j. By Observation 1, $\hat{h}_{2j}$ comprises all the positive data seen so far. Finally, $\hat{h}_{2j}$ is disjoint from the data labeled '-,' thus $2j$ passes the test, and $\hat{M}$ converges.

Case 2. j_{n_0-1} is an auxiliary hypothesis.

Now, the same argument applies *mutatis mutandis*. Either $\hat{M}$ produces in some subsequent stage an ordinary hypothesis and converges to it, or we are back to Case 1. This proves Claim A.

Finally, we have to show that the hypothesis $\hat{M}$ converges to is a correct one.

Claim B. *If $\hat{M}$ converges, say to $2k$, then $c = \hat{h}_{2k}$.*

Since $\hat{M}$ converges, we know that $2k$ is an ordinary hypothesis. Suppose $c \neq \hat{h}_k$. Let m be the first stage in which k is output. By construction, the hypothesis output in Stage $m-1$ must have been an auxiliary hypothesis, say (P, N). Let P' and N' be the corresponding finite sets $\hat{M}$ has generated before executing Instruction (β) within Stage m, and let $z + 1 = \mathrm{card}(P' \cup N')$. By $i(P', N')_z$ we denote the corresponding initial segment of some informant. Furthermore, let $r \le z$ be such that $k = M(i(P', N')_r)$. If

$i(P', N')_r$ is an initial segment of $i(c)_{y_0}$, then $r = y_0$, and we are done.

Now assume $i(P', N')_r$ to be not an initial segment of $i(c)_{y_0}$. Then $i(P', N')^+_r \not\subseteq i(c)^+_{y_0}$ or $i(P', N')^-_r \not\subseteq i(c)^-_{y_0}$. Since $\hat{M}$ has verified $P' \subseteq \hat{h}_{2k}$ and $i(P', N')^-_r \cap \hat{h}_{2k} = \emptyset$, we know that $c \neq \hat{h}_{2k}$. By Observation 1, we additionally have $P' \subseteq c$, thus we may conclude $i(P', N')^-_r \cap c \neq \emptyset$. Because of $i(P', N')^-_r \cap \hat{h}_{2k} = \emptyset$, we know that there exists an element $x \in (c \cap i(P', N')^-_r) \setminus \hat{h}_{2k}$. Taking into account that $2k$ has been first produced in Stage m, we are done if $x \in t^+_{m+\rho} \setminus t^+_m$ for some $\rho \geq 1$. Finally, assume $x \in t^+_m$. By Observation 3, the uniquely determined index μ with $x = x_\mu$ must satisfy $\mu > z$, since otherwise $x \in P'$, and $P' \subseteq \hat{h}_{2k}$ would be contradicted. However, $x \in (c \cap i(P', N')^-_r)$ yielding $\mu \leq z$ by Observation 2. This contradiction proves Claim B. ▮

The issue whether information presentation can be traded versus the mode of convergence has been treated in Lange *et al.* [17], too. In particular, in [17] we addressed the question to what extent the subset principle must be weakened if only positive data are available. The results obtained in this context imply *FIN-INF* $\subset$ *CONSV*. Because of *FB* $\subset$ *CONSV* (cf. Theorem 5, Assertion (2)), our last theorem strengthen this result.

Finally, putting the results obtained together, we obtain the already announced result that the increase in the learning power obtained by bounded example memories and feedback queries is incomparable.

THEOREM 10.

(1) $\mathit{FB} \setminus \bigcup_{k \in \mathbb{N}} \mathit{BEM}_k \neq \emptyset$,

(2) $\mathit{BEM}_{k+1} \# \mathit{FB}$ *for all* $k \in \mathbb{N}$.

Proof. Clearly, Assertion (1) follows directly from Theorem 9 and Corollary 8. Furthermore, $\mathit{BEM}_1 \setminus \mathit{FB} \neq \emptyset$ by Corollary 7, and hence, Assertion (2) follows. ▮

Next we ask what concept classes are incrementally learnable. The following subsection provides a partial answer to this problem.

3.3. *A Sufficient Condition for Incremental Learning*

There are several well-known criteria that ensure learnability in the limit of indexable classes from positive data, i.e., finite thickness and finite elasticity. Both conditions are sufficient but not necessary. Hence, it is only natural to ask whether or not these conditions guarantee any form of incremental learning, too. Unfortunately, the general answer is negative. However, a natural sharpening of these conditions directly yields sufficient conditions for all models of incremental learning introduced above.

DEFINITION 8 (Angluin [2]). *Let $\mathcal{C}$ be an indexable class. $\mathcal{C}$ has* ***finite thickness*** *iff for every $x \in \mathcal{X}$ there are at most finitely many $c \in \mathcal{C}$ satisfying $x \in c$.*

PROPOSITION 2. *There is an indexable class $\mathcal{C} \notin$ CONSV which has finite thickness.*

Proof. Consider the following indexable class $\mathcal{C}$ of languages $L_{\langle k,j \rangle}$ defined as follows. For all $k \in \mathbb{N}$, we set $L_{\langle k,0 \rangle} = \{a^k b^n \mid n \in \mathbb{N}\}$. Note that $a^0 = \varepsilon$ by convention. For all $k \in \mathbb{N}$ and all $j > 0$, we distinguish the following cases:

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

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

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

We distinguish the following subcases.

Subcase 2.1. $j < 2\Phi_k(k)$

Let $r = 2\Phi_k(k) - j$. We set $L_{\langle k,j\rangle} = \{a^k b^m \mid 1 \le m \le r\}$.

Subcase 2.2. $j \ge 2\Phi_k(k)$

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

Finally, let $\mathcal{L} = (L_{\langle k,j\rangle})_{j,k\in\mathbb{N}}$, and let $\mathcal{C} = \text{range}(\mathcal{L})$. Moreover, $\mathcal{C}$ has finite thickness by construction. On the other hand, we know that $\mathcal{C} \notin CONSV$ (cf. [15], Theorem 1). ▮

Because of Theorem 5 we may conclude:

COROLLARY 11. *There is an indexable class $\mathcal{C} \notin IT \cup FB \cup \bigcup_{k\in\mathbb{N}} BEM_k$ having finite thickness.*

Next, we define recursive finite thickness. Let $\mathcal{X}$ be any learning domain, and let $x_0, x_1, x_2, \ldots$ be any effective enumeration of all elements in $\mathcal{X}$. Furthermore, assume an effective repetition free enumeration $N_0,\ N_1,\ N_2,\ \ldots$ of all finite subsets of $\mathbb{N}$.

DEFINITION 9. *Let $\mathcal{C}$ be an indexable class. $\mathcal{C}$ has* ***recursive finite thickness*** *provided there are an indexing $c_0, c_1, c_2, \ldots$ of $\mathcal{C}$ and a total recursive function g such that, for all $m, k \in \mathbb{N}$, $x_m \in c_k$ iff $k \in N_{g(m)}$ or there is a $j \in N_{g(m)}$ with $c_j = c_k$.*

It is easy to verify that the class of all concepts describable by a monomial, a k-CNF, a k-DNF, a k-decision list, respectively have recursive finite thickness. The pattern languages provide another interesting example of a concept class having recursive finite thickness. The following theorem establishes the iterative learnability of all these concept classes.

THEOREM 12. *Let $\mathcal{C}$ be any indexable class. If $\mathcal{C}$ has recursive finite thickness, then $\mathcal{C} \in IT$.*

Proof. Let $x_0, x_1, x_2, \ldots$ be any effective enumeration of $\mathcal{X}$, let $c_0, c_1, c_2, \ldots$ be an indexing of $\mathcal{C}$, and let g be the corresponding recursive function satisfying the requirements of Definition 9. We choose the following hypothesis space $\mathcal{H} = (h_j)_{j\in\mathbb{N}}$. For every $j \in \mathbb{N}$, let $h_j = \bigcap_{k\in N_j} c_k$. Next, we define an IIM M IT–inferring $\mathcal{C}$ with respect to $\mathcal{H}$. Let $c \in \mathcal{C}$ and let $t = (s_j)_{j\in\mathbb{N}} \in pos(c)$. The IIM M is defined in stages.

Stage 0. M receives as input s_0.

Determine the least $m \in \mathbb{N}$ with $x_m = s_0$. Output $g(m)$, and go to Stage 1.

Stage n, $n \ge 1$. M receives as input its last guess, say j, and the $(n{+}1)$th element s_n of t.

Case 1. $s_n \in h_j$

Output j, and go to Stage $n+1$.

Case 2. $s_n \notin h_j$

For all $k \in N_j$, test whether or not $s_n \in c_k$. Collect all k successfully passing the test within the set $CONS$. Determine the uniquely defined index z for $CONS$, i.e., $N_z = CONS$. Output z, and go to Stage $n+1$.

Now it is straightforward to show that M iteratively infers $\mathcal{C}$ with respect to $\mathcal{H}$. We omit the details. ∎

The proof given above has some interesting features we want to point to. First, the learning algorithm produces its hypotheses in a rather *constructive* manner. This nicely contrasts the enumerative character of many inference procedures often provided in abstract studies within Gold's [11] model (cf., e.g., [5, 11, 18]). In contrast, our general learning algorithm immediately produces a finite subspace of hypotheses from which it computes its actual guess. Subsequently, it *deletes* all nonrelevant hypotheses from this subspace. Moreover, the algorithm learns by *generalization*, i.e., the sequence of its guesses constitutes an augmenting chain of concepts. As a matter of fact, the converse is also true. Whenever the learning process can be exclusively performed by generalization, then one can learn iteratively, too (cf. [14]). However, the generality of the result above does not always yield the most effective iterative learning algorithm. For example, a straightforward application of Valiant's [23] proof technique directly yields iterative learning algorithms for the class of all concepts describable by a k-CNF and k-DNF, respectively, that are much more efficient. Another example are the pattern languages. In this case, Lange and Wiehagen's [13] iterative learning algorithm is the much better choice (cf. Zeugmann [30] for a detailed analysis).

As our next result states, recursive finite thickness is only a sufficient criterion that ensures the learnability by iterative IIMs.

THEOREM 13. *There is an indexable class $\mathcal{C} \in IT$ which does not have recursive finite thickness.*

Proof. Let $\mathcal{C}_{fin}$ be the class of all finite non-empty languages over the alphabet $\{a\}$. Obviously, there are infinitely many different languages which contain the string a. Thus, $\mathcal{C}$ even does not have finite thickness. On the other hand, $\mathcal{C} \in IT$ (cf. Corollary 8). ∎

Next, we consider finite elasticity introduced by Wright [28].

DEFINITION 10. *Let $\mathcal{C}$ be an indexable class. $\mathcal{C}$ has* ***infinite elasticity*** *iff there are an infinite sequence of strings $x_0, x_1, x_2, \dots$ and an infinite sequence of concepts $c_0, c_1, c_2, \dots$ each in $\mathcal{C}$ such that, for all $n \in \mathbb{N}^+$, $\{x_0, \dots, x_{n-1}\} \subseteq c_n$ but $x_n \notin c_n$. $\mathcal{C}$ has* ***finite elasticity*** *provided that $\mathcal{C}$ does not have infinite elasticity.*

Obviously, finite thickness implies finite elasticity. Therefore, Corollary 11 yields:

COROLLARY 14. *There is an indexable class $\mathcal{C} \notin IT \cup FB \cup \bigcup_{k\in\mathbb{N}} BEM_k$ which has finite elasticity.*

On the other hand, the indexable class $\mathcal{C}_{fin}$ used in the demonstration of Theorem 13 does not have finite elasticity as well. For seeing this, set $x_j = a^{j+2}$ and $c_j = \{a^k \mid 1 \leq k \leq j+1\}$ for all $j \in \mathbb{N}$. By construction, $\{x_0, \dots, x_{n-1}\} \subseteq c_n$ but $x_n \notin c_n$. Thus $\mathcal{C}$ has infinite elasticity. Consequently:

Corollary 15. *There is an indexable class $\mathcal{C} \in IT$ which does not have finite elasticity.*

4. CONCLUSIONS AND OPEN PROBLEMS

During the last decade algorithmic learning has attracted a continuously growing interest in the computer science community. Additionally, machine learning techniques are sought after in a wider range of industrial and scientific applications, e.g., in knowledge engineering, in robotics, in pattern recognition, in financial prediction, in molecular biology, in natural language processing, and in machine discovery. Since every practical learning system has to deal with the limitations of space available, incremental learning techniques are of special interest. Moreover, it is well-known that too little information causes learning systems to fail. On the other hand, too much information may also lead to a degrading performance, a loss of efficiency, and it may even affect the accuracy. Therefore, it is of central importance to gain a better understanding of what data must be preserved during the learning process, and of what information can be overlooked. Clearly, these problems have various facets, and several of them have been studied in inductive inference (cf., e.g., Freivalds *et al.* [9] as well as Wiehagen and Zeugmann [27] and the references therein).

The present paper addresses some of these problems from a new perspective by providing a systematic study of incremental learning for indexable concept classes. Different models of incremental learning from positive data have been defined and investigated. These model differ in the way they restrict the accessibility of the input data stream. We distinguished between iterative learning, bounded example memory inference and feed-back identification.

An iterative learner is required to produce its actual hypothesis exclusively from its previous guess and the next example presented. Bounded example memory and feed-back learning generalize iterative inference by allowing to store an *a priori* bounded number of carefully chosen examples and asking whether or not a particular element did already appear in the input data provided so far, respectively.

As it turned out, all the formal models defined correspond to learning scenarios that are generally less powerful than conservative learning (cf. Theorem 5). On the other hand, by realizing a suitable interplay between the learning algorithm and the hypothesis space chosen, incremental learning may outperform conservative learning, too (cf. Theorem 1). Moreover, as the proof of Theorem 1 shows *redundancy* in the hypothesis space may seriously increase the learning capabilities of incremental learners. Future research should address the problem of what properties hypothesis spaces must have to be well suited for incremental learning.

Moreover, both feed-back learning and bounded example memory inference are more powerful than iterative learning. In particular, we rigorously proved an infinite hierarchy of more and more powerful bounded example memory learners parametrized by the number of examples storable that has been stated by Fulk *et al.* [7] (cf. Theorem 5). However, feed-back learning and bounded example memory inference extend the learning capabilities of iterative learners in different directions (cf. Theorem 10). This insight allows the conclusion that there is no unique way to design superior incremental learning algorithms.

Finally, an easy verifiable sufficient condition for incremental learning has been elaborated. Applying this criterion, the iterative learnability of all concepts describable by a monomial, a k-CNF, a k-DNF, a k-decision list, respectively, and of all the pattern languages can be shown. The importance of this criterion is mainly based on its simplicity. Once it is known that there exist an iterative learning algorithm future research can concentrate on improving its efficiency. Clearly, it would be highly desirable to have similar conditions for feed-back learning and bounded example memory inference. Ideally, one should elaborate conditions that are both necessary and sufficient for the different models of incremental learning.

ACKNOWLEDGMENTS

Most of this work has been done while the first author was visiting the Research Institute of Fundamental Information Science (RIFIS) of Kyushu University at Fukuoka, Japan. This visit has been supported by the Japanese International Information Science Foundation under Grant No. 94.3.3.543. He is greatfully indebted to Setsuo Arikawa for providing excellent working conditions during his stay at RIFIS.

Both authors heartily thank the anonymous referee, whose proposal considerable simplified the proofs of Claim 2 in Theorem 5 and Claim 3 in Theorem 6.

5. REFERENCES

1. F. Ameur, A space-bounded learning algorithm for axis-parallel rectangles, *in* "Proceedings, 2nd European Conference on Computational Learning Theory," (P. Vitanyi, Ed.), Lecture Notes in Artificial Intelligence, Vol. 904, pp. 313 – 321, Springer-Verlag, Berlin, 1995.

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

3. D. Angluin, Queries and concept learning, *Machine Learning* **2** (1988), 319 – 342.

4. M. Blum, A machine independent theory of the complexity of recursive functions, *Journal of the ACM* **14** (1967), 322 – 336.

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

6. A. Cornuéjols, Getting order independence in incremental learning, *in* "Proceedings, European Conference on Machine Learning 1993," (P.B. Brazdil, Ed.), Lecture Notes in Artificial Intelligence, Vol. 667, pp. 196 – 212, Springer-Verlag, Berlin, 1993.

7. M. Fulk, S. Jain and D. N. Osherson, Open Problems in Systems That Learn, *Journal of Computer and System Science* **49** (1994), 589 – 604.

8. E. Kinber and F. Stephan, Language learning from texts: Mind changes, limited memory and monotonicity, *in* "Proceedings, 8th Annual ACM Conference on Computational Learning Theory," (W. Maass, Ed.), pp. 182 – 189, ACM Press, New York, 1995.

9. R. Freivalds, E. Kinber and C.H. Smith, On the impact of forgetting on learning machines, *Journal of the ACM* **42** (1995), 1146 – 1168.

10. R. Godin and R. Missaoui, An incremental concept formation approach for learning from databases, *Theoretical Computer Science* **133** (1994), 387 – 419.

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

12. J.E. Hopcroft and J.D. Ullman, "Formal Languages and their Relation to Automata," Addison-Wesley, Reading, Mass., 1969.

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

14. S. Lange and T. Zeugmann, Types of monotonic language learning and their characterization, *in* "Proceedings, 5th Annual ACM Workshop on Computational Learning Theory," (D. Haussler, Ed.), pp. 377 – 390, ACM Press, New York, 1992.

15. S. Lange and T. Zeugmann, Language learning in dependence on the space of hypotheses, *in* "Proceedings, 6th Annual ACM Conference on Computational Learning Theory," (L. Pitt, Ed.), pp. 127 – 136, ACM Press, New York, 1993.

16. S. Lange and T. Zeugmann, Set-driven and rearrangement-independent learning of recursive languages, *Mathematical Systems Theory* **29**, No. 6, 1996, 599 – 634.

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

18. 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.

19. S. Porat and J.A. Feldman, Learning automata from ordered examples, *in* "Proceedings, 1st Workshop on Computational Learning Theory," (D. Haussler and L. Pitt, Eds.), pp. 386 – 396, Morgan Kaufmann Publ., San Mateo, CA, 1988.

20. R. Rivest, Learning decision lists, *Machine Learning* **2** (1987), 229 – 246.

21. T. Shinohara, Rich classes inferable from positive data: Length-bounded elementary formal systems, *Information and Computation* **108** (1994), 175 – 186.

22. L. Torgo, Controlled redundancy in incremental rule learning, *in* "Proceedings, European Conference on Machine Learning 1993," (P.B. Brazdil, Ed.), Lecture Notes in Artificial Intelligence, Vol. 667, pp. 185 – 195, Springer-Verlag, Berlin, 1993.

23. L.G. Valiant, A theory of the learnable, *Communications of the ACM* **27** (1984), 1134 – 1142.

24. K. Wexler, The subset principle is an intensional principle, *in* "Knowledge and Language: Issues in Representation and Acquisition," (E. Reuland and W. Abrahamson, Eds.), Kluwer Academic Publishers, 1992.
25. K. Wexler and P. Culicover, "Formal Principles of Language Acquisition," MIT Press, Cambridge, Mass., 1980.
26. R. Wiehagen, Limes–Erkennung rekursiver Funktionen durch spezielle Strategien, *Journal of Information Processing and Cybernetics (EIK)* **12** (1976), 93 – 99.
27. R. Wiehagen and T. Zeugmann, Learning and Consistency, *in* "Algorithmic Learning for Knowledge-Based Systems" (K.P. Jantke and S. Lange, Eds.), Lecture Notes in Artificial Intelligence, Vol. 961, pp. 1 – 24, Springer-Verlag, Berlin, 1995.
28. K. Wright, Identification of unions of languages drawn from an identifiable class, *in* "Proceedings, 2nd Annual ACM Workshop on Computational Learning Theory," (R. Rivest, D. Haussler and M.K. Warmuth, Eds.), pp. 328 – 333, Morgan Kaufmann, San Mateo, CA, 1989.
29. T. Yokomori, On polynomial-time learnability in the limit of strictly deterministic automata, *Machine Learning* **19** (1995), 153 – 179.
30. 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, April 20, 1995.
31. T. Zeugmann and S. Lange, A guided tour across the boundaries of learning recursive languages, *in* "Algorithmic Learning for Knowledge-Based Systems" (K.P. Jantke and S. Lange, Eds.), Lecture Notes in Artificial Intelligence, Vol. 961, pp. 190 – 258, Springer-Verlag, Berlin, 1995.