Language Learning in Dependence on the Space of Hypotheses

Steffen Lange*
TH Leipzig
FB Mathematik und Informatik
PF 66
O–7030 Leipzig
steffen@informatik.th-leipzig.de

Thomas Zeugmann
TH Darmstadt
Institut für Theoretische Informatik
Alexanderstr. 10
W–6100 Darmstadt
zeugmann@iti.informatik.th-darmstadt.de

Abstract

We study the learnability of indexed families $\mathcal{L} = (L_j)_{j\in\mathbb{N}}$ of uniformly recursive languages under certain monotonicity constraints. Thereby we distinguish between *exact* learnability ($\mathcal{L}$ has to be learnt with respect to the space $\mathcal{L}$ of hypotheses), *class preserving* learning ($\mathcal{L}$ has to be inferred with respect to some space $\mathcal{G}$ of hypotheses having the same range as $\mathcal{L}$), and *class comprising* inference ($\mathcal{L}$ has to be learnt with respect to some space $\mathcal{G}$ of hypotheses that has a range comprising $range(\mathcal{L})$).

In particular, it is proved that, whenever monotonicity requirements are involved, then exact learning is almost always weaker than class preserving inference which itself turns out to be almost always weaker than class comprising learning. Next, we provide additionally insight into the problem under what conditions, for example, exact and class preserving learning procedures are of equal power. Finally, we deal with the question what kind of languages has to be added to the space of hypotheses in order to obtain superior learning algorithms.

1. Introduction

Gold (1967)–style formal language learning has attracted a lot of attention during the last decades (cf. e.g. Osherson, Stob and Weinstein (1986) and the references therein). Starting with Angluin's (1980) pioneering paper many researches have focused their attention on the learnability of indexed families of uniformly recursive languages (cf. e.g. Shinohara, 1986, 1990, Kapur, 1992, Kapur and Bilardi, 1992, Lange and Zeugmann, 1992, 1993a, 1993b, Mukouchi, 1992a, 1992b). Looking at potential applications this paradigm is of special interest. Therefore we continue along this line.

The general situation investigated in language learning can be described as follows: Given more and more information concerning the language to be learnt, the inference device has to produce, from time to time, a hypothesis about the object to be inferred. The set of all admissible hypotheses is called *space of hypotheses*, or, synonymously, *hypothesis space*. The information given may contain only *positive examples*, i.e., eventually all the strings contained in the language to be learnt, as well as both *positive and negative examples*. In all what follows we deal exclusively with learning from positive data. Furthermore, the sequence of hypotheses has to converge to a hypothesis that correctly describes the target object. If there is a learning algorithm behaving as described, then the languages it learns are said to be learnable in the limit (cf. Gold, 1967). Moreover, there are many possible requirements to the sequence of all created hypotheses. In this paper we study the power of learning algorithms that, when receiving more and more information about the target language, do produce *exclusively* better and better generalizations and specializations, respectively. In its strongest formalization we require the learning algorithm to create a sequence of hypotheses describing an augmenting (descending) chain of languages, i.e., $L_i \subseteq L_j$ ($L_i \supseteq L_j$) iff L_j is later guessed than L_i (cf. Definition 3, (A), (B)). Following Jantke (1991) and Kapur (1992) we call these modes of inference strong–monotonic and dual strong–monotonic. Weakening the latter demands has led to the notion of weak–monotonic and dual weak–monotonic learning. Here an inference machine is obligated to behave itself strong–monotonically (dual strong–monotonically) as long as it receives data not contradicting its actual hypothesis. If it gets data that provably misclassify its last hypothesis, then it is allowed to output any new guess (cf. Definition 3, (C), (D)).

Our main goal consists in investigating the *learning power in dependence on the choice of the space of hypotheses.* For the sake of presentation we introduce some notations. An *indexed family* $\mathcal{L} = (L_j)_{j\in\mathbb{N}}$ is a recursive

*This research has been supported by the German Ministry for Research and Technology (BMFT) under grant no. 01 IW 101.

enumeration of non–empty languages such that membership in L_j is uniformly decidable for all $j \in \mathbb{N}$. If an indexed family $\mathcal{L}$ can be learnt with respect to $\mathcal{L}$ itself, then $\mathcal{L}$ is said to be *exactly* learnable. That means, the learning algorithm uses $\mathcal{L}$ as hypothesis space. Furthermore, $\mathcal{L}$ is learnable by a *class preserving* learning algorithm M, if there is a space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ of hypotheses such that any G_j describes a language from $\mathcal{L}$ and M infers $\mathcal{L}$ with respect to $\mathcal{G}$. Now any produced hypothesis is required to describe a language belonging to $\mathcal{L}$ but we are free to use a possibly *different enumeration* of $\mathcal{L}$ and possibly *different descriptions* of any $L \in \mathcal{L}$. Finally, we consider *class comprising* learning. In this setting a learning algorithm is allowed to use any hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ such that any $L \in \mathcal{L}$ possesses a description G_j but $\mathcal{G}$ may additionally contain elements G_k not describing any language from $\mathcal{L}$. Since membership in $\mathcal{L}$ is uniformly decidable, we restrict ourselves to consider exclusively spaces of hypotheses having a uniformly decidable membership problem.

Historically, most authors have investigated exact learnability. Moreover, many investigations in other domains of algorithmic learning theory deal with exact learning too (cf. e.g. Natarajan, 1991). And indeed, as long as one considers learning in the limit without any additional demand, every indexed family being learnable class comprisingly may be learnt exactly too (cf. Lange and Zeugmann, 1993b, 1993c). However, when dealing with characterizations it turned out to be very helpful to construct class preserving spaces of hypotheses (cf. Kapur and Bilardi, 1992), Lange and Zeugmann, 1992). Consequently, it is only natural to ask whether or not class preserving learning algorithms are more powerful than exact ones. Dealing with a measure of efficiency we found that an appropriate choice of the hypothesis space may eventually increase the learning power (cf. Lange and Zeugmann, 1993b, 1993c). Furthermore, studying the capabilities of learning algorithms in dependence on the hypothesis space has yield very interesting results concerning probabilistic learning models (cf. eg. Anthony and Gibbs, 1992, Freivalds, Kinber and Wiehagen, 1988). Therefore, it is worth to study this phenomenon in some more detail.

First, we present results demonstrating the superiority of class comprising to class preserving monotonic learning algorithms that are themselves superior to exact ones. These separations have been obtained by developing a *new powerful proof technique*. Establishing the announced separations using standard proof techniques would require to diagonalize against all spaces of hypotheses and all learning algorithms. Instead, we have elaborated an *effective reduction* of the halting problem to monotonic learning problems. This approach yields easy to describe indexed families witnessing the desired separations.

Next, we ask why, for example, class preserving inference procedures are sometimes more powerful than exact learning algorithms. Obviously, as long as there is an effective compiler from the space $\mathcal{G}$ of hypotheses into the indexed family $\mathcal{L}$, both modes of inference are of equal power. Looking at learning in the limit, Gold (1967) proved that even limiting recursive compilers do suffice. What we would like to present are characterization results stating that exact learning is of the same power than class preserving inference if and only if there are limiting recursive compilers satisfying appropriate monotonicity requirements (cf. Theorem 4 and 5). Hence, our separations prove the non–existence of such compilers.

Finally, we present results comparing class comprising and exact inference procedures. These results strongly recommend the designer of learning algorithms to carefully choose the enumeration as well as the description of the languages to be inferred in order to obtain exact learning procedures of maximal power (cf. Theorem 5 and 7). These results have been obtained using non–trivial generalizations of our theorems that characterize monotonic inference in terms of finitely generable recursive sets (cf. Lange and Zeugmann, 1992).

The paper is structured as follows. Section 2 presents preliminaries. Separations are established in Section 3. Results dealing with the announced characterizations in terms of limiting recursive compilers are presented in Section 4. The construction of description languages yielding superior learning algorithms is given in Section 5. Conclusions and open problems are outlined in Section 6, while Section 7 comprises all references.

2. Preliminaries

By $\mathbb{N} = \{1, 2, 3, ...\}$ we denote the set of all natural numbers. In the sequel we assume familiarity with formal language theory. By Σ we denote any fixed finite alphabet of symbols. Let Σ^* be the free monoid over Σ. Any subset $L \subseteq \Sigma^*$ is called a language. By $co - L$ we denote the complement of L. Let L be a language and $t = s_1, s_2, s_3, ...$ an infinite sequence of strings from Σ^* such that $range(t) = \{s_k \mid k \in \mathbb{N}\} = L$. Then t is said to be a *text* for L or, synonymously, a *positive presentation*. Let L be a language. By $text(L)$ we denote the set of all positive presentations of L. Moreover, let t be a text and let x be a number. Then t_x, denote the initial segment of t of length x. Let t be a text and let $x \in \mathbb{N}$. We set $t_x^+ := \{s_k \mid k \leq x\}$.

Next, we introduce the notion of *canonical text* that turned out to be very helpful in proving several theorems. Let L be any non–empty recursive language, and let $s_1, s_2, s_3, ...$ be the lexicographically ordered text of Σ^*. The canonical text of L is obtained as fopllows. Test sequentially whether $s_z \in L$ for $z = 1, 2, 3, ...$ until the first z is found such that $s_z \in L$. Since $L \neq \emptyset$ there must be at least one z fulfilling the test. Set $t_1 = s_z$. We proceed inductively:

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

In the sequel we deal with the learnability of indexed families of uniformly recursive languages defined as follows (cf. Angluin, 1980):
A sequence $L_1, L_2, L_3, ...$ is said to be an *indexed family* $\mathcal{L}$ of uniformly recursive languages provided all L_j are non–empty and there is a recursive function f such that for all numbers j and all strings $s \in \Sigma^*$ we have

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

In all what follows we refer to indexed families of uniformly recursive languages as indexed families for short. Moreover, we often denote an indexed family and its range by the same symbol $\mathcal{L}$. What is meant will be clear from the context.

As in Gold (1967) we define an *inductive inference machine* (abbr. IIM) to be an algorithmic device which works as follows: The IIM takes as its input larger and larger initial segments of a text t and it either requires the next input string, or it first outputs a hypothesis, i.e., a number encoding a certain computer program, and then it requires the next input string.

At this point we have to clarify what spaces of hypotheses we should choose. We require the inductive inference machines to output grammars, since this learning goal fits well with the intuitive idea of language learning. Furthermore, since we exclusively deal with indexed families $\mathcal{L} = (L_j)_{j \in \mathbb{N}}$ we always take as space of hypotheses an enumerable family of grammars $G_1, G_2, G_3, ...$ over the terminal alphabet Σ satisfying $\mathcal{L} \subseteq \{L(G_j) \mid j \in \mathbb{N}\}$. Moreover, we require that membership in $L(G_j)$ is uniformly decidable for all $j \in \mathbb{N}$ and all strings $s \in \Sigma^*$. The IIM outputs numbers j which we interpret as G_j.

A sequence $(j_x)_{x \in \mathbb{N}}$ of numbers is said to be convergent in the limit iff there is a number j such that $j_x = j$ for almost all numbers x.

Definition 1. (Gold, 1967)

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

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

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

Note that, in general it is undecidable whether or not an IIM has already successfully finished its learning task. If this decidability is additionally required, then we obtain *finite* learning.

Definition 2. (Gold, 1967)

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

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

The next definition formalizes the notions of monotonicity that we have informally introduced in the introduction.

Definition 3. (Jantke, 1991, Kapur, 1992)

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

(A) strong–monotonically

(B) dual strong–monotonically

(C) weak–monotonically

(D) dual weak–monotonically

iff

M LIM–identifies L and for any text $t \in text(L)$ as well as for any two consecutive hypotheses j_x, j_{x+k} which M has produced when fed t_x and t_{x+k} for some $k \geq 1, k \in N$, the following conditions are satisfied:

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

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

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

(D) *if* $t_{x+k} \subseteq L(G_{j_x})$ *then* $co{-}L(G_{j_x}) \subseteq co{-}L(G_{j_{x+k}})$.

By $SMON$, $SMON^d$, $WMON$, and $WMON^d$ we denote the family of all sets $\mathcal{L}$ of indexed families for which there is an IIM inferring it strong–monotonically, dual strong–monotonically, weak–monotonically, and dual weak–monotonically, respectively.

Note that weak–monotonic inference equals conservative learning originally introduced by Angluin (1980) (cf. Lange and Zeugmann, 1993a).

For any mode of inference defined above we use the prefix E to denote exact learning, i.e., the fact that $\mathcal{L}$ has to be inferred with respect to $\mathcal{L}$ itself. For example, $ELIM$ denotes exact learnability in the limit. Moreover, we use the prefix C to mark class comprising inference, i.e., the fact that $\mathcal{L}$ may be learnt with respect to any space of hypotheses comprising $range(\mathcal{L})$. To have an example, $CSMON^d$ denotes the collection of all indexed families that can be class comprisingly learnt by a dual strong–monotonically working IIM. If no prefix is used, then we always mean class preserving learning, i.e., learning with respect to a space $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ of hypotheses such that $range(\mathcal{L}) = \{L(G_j) \mid j \in \mathbb{N}\}$.

In the next section we compare the learning power of the introduced modes of inference in dependence on the space of hypotheses.

3. Separations

The starting point of our investigations has been Angluin's (1980) theorem stating $EWMON \subset ELIM$. In essence, this theorem asserts that an IIM sometimes has to reject a guess *without having received data contradicting it.* A careful analysis of her proof even showed that any indexed family $\mathcal{L} \in ELIM \setminus EWMON$ cannot be learnt by an IIM that works semantically finite. An IIM is said to work *semantically finite* when learning $\mathcal{L}$ with respect to $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ if and only if for all $L \in \mathcal{L}$ any any text $t \in text(L)$ the following condition is satisfied: Let j be the hypothesis the sequence $(M(t_x))_{x\in\mathbb{N}}$ converges to and let z be the least number such that $M(t_z) = j$. Then, $L(G_{M(t_y)}) \neq L(G_j)$ for all $y < z$. That means, a semantically finite working IIM is never allowed to reject a guess which is correct for the language to be learnt.

Hence, it is only natural to ask why $EWMON \subset ELIM$. Looking at inductive inference of recursive functions we have, roughly speaking, the following situation. Any IIM M inferring in the limit some class U of recursive functions may be replaced by a semantically finite working IIM $\hat{M}$ that learns U as well. This result is achieved by constructing a suitable space of hypotheses comprising U which respect to which the desired IIM $\hat{M}$ may infer U semantically finite (cf. Wiehagen, 1991).

Hence, we first conjectured that $EWMON \subset ELIM$ might be caused by the requirement to learn exactly. Subsequently, we observed that $ELIM = LIM = CLIM$ as well as $EWMON \subset WMON$ (cf. Lange and Zeugmann, 1993b, 1993c). Next we have shown that $CSMON \setminus WMON \neq \emptyset$ but, however, $CWMON \subset LIM$. In particular, the latter statement shows that, when learning from positive data, non–semantically finite working IIMs are sometimes inevitable. Moreover, the results mentioned stimulated us to investigate the power of learning in dependence on the space of hypotheses in detail. We have obtained the following figure.

Theorem 1.

$$\begin{array}{ccccc}
ELIM & = & LIM & = & CLIM \\
\cup & & \cup & & \| \\
EWMON^d & \subset & WMON^d & \subset & CWMON^d \\
\cup & & \cup & & \cup \\
EWMON & \subset & WMON & \subset & CWMON \\
\cup & & \cup & & \cup \\
ESMON & \subset & SMON & \subset & CSMON \\
\cup & & \cup & & \cup \\
EFIN & = & FIN & = & CFIN \\
\| & & \| & & \cap \\
ESMON^d & = & SMON^d & \subset & CSMON^d
\end{array}$$

Proof. All improper inclusions immediately follow from the definitions of the corresponding identification types. The equality of $ELIM$, LIM and $CLIM$ has been proved in Lange and Zeugmann (1993c). For the proof of $EFIN = FIN = CFIN$ the reader is referred to Lange and Zeugmann (1993d).

Now we show $EWMON^d \setminus CWMON \neq \emptyset$. This implies $\lambda WMON \subset \lambda WMON^d$, for any $\lambda \in \{E, \epsilon, C\}$, where ϵ denotes the empty string.

Let φ_1, φ_2, φ_3,... be any fixed acceptable programming system for the partial recursive functions over $\mathbb{N}$, and let Φ_1, Φ_2, Φ_3,... be any associated complexity measure (cf. Machtey and Young, 1978). By $c : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ we denote Cantor's pairing function.

The desired indexed family $\mathcal{L}_{ewd}$ is defined as follows. For all $k \in \mathbb{N}$ we set $L_{c(k,1)} = \{a^k b^n \mid n \in \mathbb{N}\}$. For all $k \in \mathbb{N}$ and all $j > 1$ we distinguish the following cases:

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

Then we set $L_{c(k,j)} = L_{c(k,1)}$.

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

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

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

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

Claim A. $\mathcal{L}_{ewd} \notin CWMON$

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

Lemma. *If there exists an IIM M with $\mathcal{L}_{ewd} \subseteq CWMON(M)$, then one can effectively construct an algorithm deciding for all $k \in \mathbb{N}$ whether or not $\varphi_k(k)$ converges.*

Proof. Let M be any IIM weak–monotonically inferring $\mathcal{L}_{ewd}$ w.r.t. some hypothesis space $\mathcal{G}$ comprising $\mathcal{L}_{ewd}$. We define an algorithm $\mathcal{A}$ that solves the halting problem. On input $k \in \mathbb{N}$ the algorithm $\mathcal{A}$ executes the following instructions:

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

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

Since M has to infer $L_{c(k,1)}$ in particular from t, there has to be a least z such M on input t_z computes a hypothesis j satisfying $t_z^+ \cup \{a^k b^{z+1}\} \subseteq L(G_j)$. Moreover, the test whether or not $t_z^+ \cup \{a^k b^{z+1}\} \subseteq L(G_j)$ can be effectively performed, since membership in $L(G_j)$ is uniformly decidable. By the definition of a complexity measure, instruction (A2) is effectively executable. Hence, $\mathcal{A}$ is an algorithm.

It remains to show that $\varphi_k(k)$ diverges, if $\neg\ \Phi_k(k) \leq z$. Suppose the converse; then there exists a $y > z$ with $\Phi_k(k) = y$. In accordance with the definition of $\mathcal{L}_{ewd}$, there is an $L \in \mathcal{L}_{ewd}$, namely $L = t_z^+$, such that $L \subset L(G_j)$. Since $t_z^+ = L$, we know that t_z is also an initial segment of a text $\hat{t}$ for L. Hence, M does not work weak–monotonically when inferring L on $\hat{t}$. This contradicts our assumption that $\mathcal{L}_{ewd} \subseteq CWMON(M)$. Hence, the lemma is proved.

Claim B. $\mathcal{L}_{ewd} \in EWMON^d$
We have to show that there is an IIM M that infers $\mathcal{L}_{ewd}$ dual strong–monotonically w.r.t. $\mathcal{L}_{ewd}$ itself.

Let $L \in \mathcal{L}_{ewd}$, let $t \in text(L)$, and let $x \in \mathbb{N}$. We define:

$M(t_x) =$ "Determine the unique k such that $t_1 = a^k b^m$ for some $m \in \mathbb{N}$. Test whether or not $\Phi_k(k) \leq x$. In case it is, goto (A). Otherwise, output $c(k,1)$.

(A) Test whether or not $a^k b^{\Phi_k(k)+m} \in t_x^+$ for some $m \in \mathbb{N}$. In case it is, output $c(k,1)$. Otherwise, goto (B).

(B) Determine the maximal $m \in \mathbb{N}$ such that $a^k b^m \in t_x^+$. Output $c(k, 2 \cdot \Phi_k(k) - m)$."

By construction, M performs at most one mind change which is not caused by an inconsistency when inferring L from text t. This is, if ever, a mind change from $L_{c(k,1)}$ to one of its finite sublanguages contained in $\mathcal{L}_{ewd}$. All remaining mind changes are justified ones. Therefore, M works dual weak–monotonically. Moreover, for any $k \in \mathbb{N}$ there are at most finitely many different finite languages $L_{c(k,j)} \subset L_{c(k,1)}$. Hence, M converges to a correct hypothesis for L. This proves Claim B.

$EWMON^d \subset ELIM$ as well as $CWMON^d = LIM$ has been shown in Lange, Zeugmann and Kapur (1992). The family witnessing $ELIM \setminus EWMON^d \neq \emptyset$ can also be used to prove $EWMON^d \subset WMON^d$, (cf. Lange and Zeugmann, 1993d). The inclusion $WMON^d \subset LIM$ is due to E.B. Kinber (1993).

Since $CSMON \setminus WMON \neq \emptyset$ (cf. assertion (1) of Theorem 3 below), one directly obtains $WMON \subset CWMON$ and $SMON \subset CSMON$. Similarly, $SMON \setminus EWMON \neq \emptyset$ (cf. assertion (3) of Theorem 3) implies $EWMON \subset WMON$ as well as $ESMON \subset SMON$. Furthermore, since $EWMON \setminus CSMON \neq \emptyset$ (cf. assertion (2) of Theorem 3), we have $\lambda SMON \subset \lambda WMON$, for any $\lambda \in \{E, \epsilon, C\}$.

Moreover, assertion (4) of Theorem 3 yields $CFIN \subset CSMON$, $CFIN \subset CSMON$, and $SMON^d \subset CSMON^d$. $FIN = SMON^d$ has been shown in Lange, Zeugmann and Kapur (1992). Finally, $ESMON \setminus FIN \neq \emptyset$ may be demonstrated using the indexed family $\mathcal{L} = (L_k)_{k \in \mathbb{N}}$, where $L_k = \{a^n \mid n \leq k\}$. This completes the proof of Theorem 1.
q.e.d.

In Lange and Zeugmann (1993a) we have shown that strong–monotonic inference from positive and negative data is always less powerful than weak–monotonic learning from text. On the other hand, $SMON \setminus EWMON \neq \emptyset$ and $CSMON \setminus WMON \neq \emptyset$ as we shall see below. Consequently, it is only natural to ask what the relation between $CSMON^d$ and $CWMON$ is. We have been a bit surprised to obtain the next theorem.

Theorem 2. $CSMON^d \subset EWMON$

Proof. We start with the following lemma.

Lemma. *Let $\mathcal{L} = (L_j)_{j \in \mathbb{N}}$ be an indexed family. If $\mathcal{L} \in CSMON^d$, then either $L_j = L_k$ or $L_k \# L_j$ for all $j,\ k \in \mathbb{N}$.*

Proof. Suppose the converse, i.e., there is an indexed family $\mathcal{L} \in CSMON^d$ such that $L_j \subset L_k$ for some $j,\ k \in \mathbb{N}$. Let M denote any IIM inferring $\mathcal{L}$ dual strong–monotonically w.r.t. to a hypothesis space $\mathcal{G}$ comprising $\mathcal{L}$. Moreover, let $t \in text(L_j)$. Since M has to infer L_j from t, there has to be an $x \in \mathbb{N}$ such that $M(t_{x+r}) = j_x$ for all $r \in \mathbb{N}$ and $L(G_{j_x}) = L_j$. Since $L_j \subset L_k$, t_x can be extended to a text $\hat{t}$ for L_k. It is easy to see that M fails to infer L_k from $\hat{t}$ dual strong-monotonically. This proves the lemma.

Claim A. $CSMON^d \subseteq EWMON$.

Let $\mathcal{L} \in CSMON^d$. The wanted IIM M is defined as follows. Let $L \in \mathcal{L}$, let $t \in text(L)$, and let $x \in \mathbb{N}^+$. We set:

$M(t_x) =$ "Search the least index $j \leq x$ with $t_x^+ \subseteq L_j$. Output it. In case $t_x^+ \not\subseteq L_j$ for all $j \leq x$ output nothing and request the next input."

By construction, M performs exclusively justified mind changes. Let $z = \mu j[L = L_j]$. By the lemma above we may conclude that that $L_z \setminus L_k \neq \emptyset$ for all $k < z$. Consequently, for all $k < z$ there has to be an x_k such that $t_{x_k}^+ \not\subseteq L_k$. Therefore, M rejects any hypothesis k with $k < z$ and converges to z. This proves Claim A.

Claim B. $EWMON \setminus CSMON^d \neq \emptyset$.
Consider the indexed family $\mathcal{L}$ with $L_k = \{a^n \mid n \leq k\}$. It is straightforward to show $\mathcal{L} \in ESMON$, and hence $\mathcal{L} \in EWMON$. On the other hand, by the lemma above one immediately gets $\mathcal{L} \notin CSMON^d$.
q.e.d.

Theorem 3.

(1) $WMON \ \# \ CSMON$

(2) $EWMON \ \# \ CSMON$

(3) $EWMON \ \# \ SMON$

(4) $CSMON^d \ \# \ CSMON$

Proof. We start with assertion (4). Within the proof of Theorem 2 we have already shown that $ESMON \setminus CSMON^d \neq \emptyset$. This directly implies $CSMON \setminus CSMON^d \neq \emptyset$. Hence, it suffices to define an indexed family $\mathcal{L}_{csd}$ such that $\mathcal{L}_{csd} \in CSMON^d$, but $\mathcal{L}_{csd} \notin CSMON$.

We use the notations introduced in the proof of Theorem 1.

The desired indexed family is defined as follows. For all $k \in \mathbb{N}$ we set $L_{c(k,1)} = \{a^k b^n \mid n \in \mathbb{N}\}$. For all $k \in \mathbb{N}$ and all $j > 1$ we distinguish the following cases:

Case 1: $\neg\ \Phi_k(k) \leq j$
We set: $L_{c(k,j)} = L_{c(k,1)}$

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

$$L_{c(k,j)} = \begin{cases} \{a^k b^m \mid m \leq d\} \cup \{b^k a^j\}, \text{ if } d \geq 1 \\ \{a^k b\} \cup \{b^k a^j\}, \qquad \text{otherwise} \end{cases}$$

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

Claim A. $\mathcal{L}_{csd} \notin CSMON$

The claim is proved by reducing the halting problem to $\mathcal{L}_{csd} \in CSMON$. We omit the details.

Claim B. $\mathcal{L}_{csd} \in CSMON^d$

We have to show that there is an appropriate space of hypotheses $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ comprising $\mathcal{L}_{csd}$ and an IIM M inferring $\mathcal{L}$ dual strong–monotonically w.r.t. $\mathcal{G}$.

We define the wanted space of hypotheses $\mathcal{G}$ as follows. For all k, $j \in \mathbb{N}$, we set $L(G_{c(k,1)}) = \bigcup_{j \in \mathbb{N}} L_{c(k,j)}$ and $L(G_{c(k,j+1)}) = L_{c(k,j)}$.

Since $\mathcal{L}_{csd}$ is an indexed family, it is easy to verify that membership is uniformly decidable for $\mathcal{G}$. By construction, $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ comprises $\mathcal{L}_{csd}$. For the proof of $\mathcal{L}_{csd} \in CSMON^d$ w.r.t. $\mathcal{G}$ the reader is referred to Lange and Zeugmann (1993d).

The proof idea used above applies mutatis mutandis to obtain the remaining separations. Next to, we separate $SMON$ and $EWMON$ which completes the proof of assertion (3).

The desired indexed family $\mathcal{L}_{cs}$ is defined as follows. For all $k \in \mathbb{N}$, we set $L_{c(k,1)} = \{a^k b^n \mid n \in \mathbb{N}\}$. For all $k \in \mathbb{N}$ and all $j > 1$, we distinguish the following cases:

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

We set: $L_{c(k,j)} = L_{c(k,1)}$

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

Then, we set: $L_{c(k,j)} = \{a^k b^m \mid m \leq \Phi_k(k)\}$

The proof of $\mathcal{L}_{cs} \in SMON \setminus EWMON$ may be found in Lange and Zeugmann (1993d).

In order to prove assertion (1) and assertion (2), it remains to show $CSMON \setminus WMON \neq \emptyset$. This can be done using the following indexed family $\mathcal{L}_{csmon}$. For all $k \in \mathbb{N}$ we set $L_{c(k,1)} = \{a^k b^n \mid n \in \mathbb{N}\}$. For all $k \in \mathbb{N}$ and all $j > 1$ we distinguish the following cases:

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

We set: $L_{c(k,j)} = L_{c(k,1)}$

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

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

$L_{c(k,j)} = \{a^k b^m \mid m \leq \Phi_k(k)\} \cup \{a^k b^{\Phi_k(k)+2(d+m)} \mid m \in \mathbb{N}\}$

We omit the details.

q.e.d.

As we have seen, a new proof technique has been applied to obtain most of the stated separations. In particular, we have effectively reduced the halting problem to several monotonic learning problems. These reductions imply that the considered learning problems are at least as hard the halting problem. This insight deserves some attention. Gold (1967) showed that that no IIM can learn the class $\mathcal{R}$ of all recursive functions in the limit. On the other hand, the degree of the algorithmic unsolvability of $\mathcal{R} \in LIM$ is strictly less than the degree of the halting problem (cf., Adleman and Blum, 1991). This puts the constraint to learn monotonically w.r.t. a particular hypothesis space into a new perspective. An algorithmically solvable learning problem (e.g. $\mathcal{L} \in CSMON$) may become algorithmically unsolvable, if an at first glance natural demand is added (e.g. to learn exactly), even if the monotonicity requirement is slightly weakened (cf., e.g. assertion (2) of Theorem 3). Moreover, the degree of unsolvability may be at least as high as that of the halting problem, and is, therefore, strictly higher than that of learning all recursive functions. As far as we know, there is only one paper stating an analogous result in the setting of inductive inference of recursive functions, i.e., Freivalds, Kinber and Wiehagen (1992).

In the next section we provide some more insight into the problem why the choice of the space of hypotheses does considerably influence the power of monotonic learning algorithms.

4. Limiting Recursive Compilers

This section is devoted to the problem why an indexed family that can be learnt with respect to some space $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ of hypotheses might become non–inferable with respect to other spaces $\hat{\mathcal{G}} = (\hat{G}_j)_{j \in \mathbb{N}}$ of hypotheses. A first hint how to answer this question has already been given by Gold (1967). Namely, he proved that, whenever there is a limiting recursive compiler (cf. Definition 4 below) ¿from $\mathcal{G}$ into $\hat{\mathcal{G}}$, then any IIM inferring a class $\mathcal{L}$ of languages with respect to $\mathcal{G}$ can easily be converted into one that learns $\mathcal{L}$ with respect to $\hat{\mathcal{G}}$. Considering indexed families being learnable with respect to some space $\mathcal{G}$ of hypotheses, we could prove that there is always a limiting recursive compiler from $\mathcal{G}$ into $\mathcal{L}$. The same is, mutatis mutandis, true for finite learning, i.e., there is always a recursive compiler from $\mathcal{G}$ into $\mathcal{L}$. However, if some monotonicity requirement is involved, then the situation considerably changes. The reason for that phenomenon is as follows. A limiting recursive compiler in general does not preserve any of the introduced monotonicity demands. But even if it does, it is a highly non–trivial task to convert an IIM that, for example, class preservingly learns an indexed family $\mathcal{L}$ with respect to some appropriate chosen space $\mathcal{G}$ of hypotheses into an IIM exactly learning $\mathcal{L}$. The latter difficulty is caused by the fact that one has to combine two limiting processes into one. We have solved this problem by using a suitable modification of our characterization theorems (cf. Lange and Zeugmann, 1992).

For the sake of presentation we give only two of the theorems obtained, since they already do suffice to convey the spirit of the insight achievable. We start with the formal definition of limiting recursive compilers. For notational convenience we use $L(\mathcal{G})$ to denote $\{L(G_j) \mid j \in \mathbb{N}\}$ for any space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ of hypotheses.

Definition 4. *Let $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ and $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ be two spaces of hypotheses such that $L(\mathcal{G}) = L(\hat{\mathcal{G}})$. A recursive function $f : \mathbb{N}\times\mathbb{N} \to \mathbb{N}$ is said to be a limiting recursive compiler from $\mathcal{G}$ into $\hat{\mathcal{G}}$ iff $k := \lim_{x\to\infty} f(j,x)$ exists and satisfies $L(G_j) = L(\hat{G}_k)$ for all $j \in \mathbb{N}$.*

Next to we introduce limiting recursive compilers fulfilling certain monotonicity demands.

Definition 5. *Let $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ and $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ be two spaces of hypotheses such that $L(\mathcal{G}) = L(\hat{\mathcal{G}})$. A limiting recursive compiler f from $\mathcal{G}$ into $\hat{\mathcal{G}}$ is said to be*

(A) strong–monotonic

(B) weak–monotonic

iff

(A) for all $j, x \in \mathbb{N}$ we have $L(\hat{G}_{f(j,x)}) \subseteq L(\hat{G}_{f(j,x+1)})$

(B) for all $j, x \in \mathbb{N}$ we have, if $k = \lim_{x\to\infty} f(j,x)$, then $L(\hat{G}_{f(j,x)}) \not\supset L(G_k)$.

Now we are ready to present the announced characterizations comparing the power of exact and class preserving learning algorithms under certain monotonicity constraints.

Theorem 4. *Let $\mathcal{L}$ be an indexed family and let $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ be a space of hypotheses such that $\mathcal{L} \in SMON$ w.r.t. $\mathcal{G}$. Then we have:*

$\mathcal{L} \in ESMON$ if and only if there is a strong–monotonic limiting recursive compiler from $\mathcal{G}$ into $\mathcal{L}$.

Proof. Necessity. Let $\mathcal{L} \in ESMON$. Then there is an IIM M such that $\mathcal{L} \subseteq ESMON(M)$ with respect to $\mathcal{L}$. We define the desired limiting recursive compiler from $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ into $\mathcal{L}$ as follows. Let $j, x \in \mathbb{N}$ and let t^j be the canonical text of $L(G_j)$. We set:

$f(j,x) =$ "Compute the sequence $(M(t^j_z))_{z\in\mathbb{N}}$ up to length x. Let j_y be the last element of this sequence. Set $f(j,x) = j_y$."

It is straightforward to verify that f is strong–monotonic limiting recursive compiler from $\mathcal{G}$ into $\mathcal{L}$.

Sufficiency. Let $\mathcal{L} \subseteq SMON(\hat{M})$ w.r.t. $\mathcal{G}$ and let f be a strong–monotonic limiting recursive compiler from $\mathcal{G}$ into $\mathcal{L}$. We have to define an IIM such that $\mathcal{L} \subseteq ESMON(M)$. The main difficulty we have to deal with is the combination of two limiting recursive processes into one yielding an IIM strong–monotonically inferring $\mathcal{L}$ w.r.t. $\mathcal{L}$. Let t be any text of any language $L \in \mathcal{L}$. Furthermore, let $j_x = \hat{M}(t_x)$ and $j_{x+r} = \hat{M}(t_{x+r})$. Then, of course, we have $L(G_{j_x}) \subseteq L(G_{j_{x+r}})$. But it is by no means obvious whether or not $L_{f(j_x,z)} \subseteq L_{f(j_{x+r},m)}$ or how to choose z and m such that the resulting hypotheses do fulfil the desired monotonicity demands. However, a suitable modification of our characterization of $SMON$ in terms of *finite tell-tales* offers a possibility to handle the inclusion problem recursively. Therefore, we continue with the following lemmata.

Lemma 1. *There is a recursively generable family $(T^y_j)_{j,y\in\mathbb{N}}$ of finite sets defined with respect to the hypothesis space $\mathcal{G}$ such that the following conditions are fulfilled:*

(1) for all $L \in \mathcal{L}$ there is a j with $L = L(G_j)$ and $T^y_j \neq \emptyset$ for almost all $y \in \mathbb{N}$,

(2) for all $j, y \in \mathbb{N}$ we have: $T^y_j \subseteq L(G_j)$ and $T^y_j \neq \emptyset$ implies $T^{y+1}_j = T^y_j$,

(3) for all $j, y, z \in \mathbb{N}$ we have: $\emptyset \neq T^y_j \subseteq L_z$ and $\emptyset \neq T^y_j \subseteq L(G_z)$ implies $L(G_j) \subseteq L_z$ and $L(G_j) \subseteq L(G_z)$, respectively.

We define the wanted sets T^y_j as follows. Let $j, y \in \mathbb{N}$; then we set:

$$T^y_j = \begin{cases} Val\, t^j_z, & \text{if } \; z = min\{x \mid x \leq y, \\ & \qquad \hat{M}(t^j_x) = j\}, \\ \emptyset, & \text{otherwise.} \end{cases}$$

where $\hat{M}$ denotes the IIM that strong–monotonically infers $\mathcal{L}$ w.r.t. $\mathcal{G}$. Using mutatis mutandis the same arguments as in the proof of Theorem 2 in Lange and Zeugmann (1992) one easily verifies that assertion (1), (2) and (3) are satisfied.
This proves Lemma 1.

Lemma 2. *Let $(T^y_j)_{j,y\in\mathbb{N}}$ be the family of recursively generable finite sets from Lemma 1. Then there is an IIM $\tilde{M}$ satisfying the following properties:*

(1) $\mathcal{L} \subseteq SMON(\tilde{M})$,

(2) for all $L \in \mathcal{L}$ and all $t \in Text(L)$ we have: $z = \lim_{x\to\infty} \tilde{M}(t_x)$ implies $T^y_z \neq \emptyset$ for almost all $y \in \mathbb{N}$.

We define the desired IIM $\tilde{M}$ as follows:

$\tilde{M}(t_x) =$ "For $j = 1, ..., x$, generate T^x_j and test for all non-empty T^x_j whether or not $T^x_j \subseteq t^+_x \subseteq L(G_j)$. In case there is at least a j fulfilling the test, output the minimal one and request the next input.

Otherwise output nothing and request the next input."

It remains to show that assertion (1) and (2) are satisfied.

Claim. $\tilde{M}$ infers $\mathcal{L}$ w.r.t. $\mathcal{G}$.

Let $k = \mu z[L = L(G_z)$ and $T^y_z \neq \emptyset$ for almost all $y]$. Assertion (1) of Lemma 1 ensures the existence of such a k. Furthermore, by assertion (2) of Lemma 1 there is a y_0 satisfying $T^0_k = ... = T^{y_0}_k = \emptyset$ and $\emptyset \neq T^{y_0+1}_k = T^{y_0+r}_k$ for all $r \in \mathbb{N}$.

The latter observation also holds for all j for which there is a y such that $T^y_j \neq \emptyset$. Consequently, any rejected hypothesis is never repeated in some subsequent step. The remaining part of (1) may be analogously shown as in the proof of Theorem 2 in Lange and Zeugmann (1992).

Assertion (2) is an immediate consequence of $\tilde{M}$'s definition and the observation made in proving the claim above.
This proves Lemma 2.

We continue with the definition of the desired IIM M. Let $L \in \mathcal{L}$, $t \in text(L)$ and $x \in \mathbb{N}$. Moreover, let $(T_j^y)_{j,y \in \mathbb{N}}$ be the recursively generable family of Lemma 1, and let $\tilde{M}$ be chosen in accordance with Lemma 2. We set:

$M(t_x) =$ "If $x = 0$ or M, when having successively fed with t_{x-1} does not output any guess, then goto (A). Else goto (B).

(A) Simulate $\tilde{M}$ on input t_x. If $\tilde{M}$, when fed successively with t_x, does not produce any hypothesis, then output nothing and request the next input.
Otherwise, let $j = \tilde{M}(t_x)$. Set $FLAG(x) = j$, output $f(j, x)$ and request the next input.
(* By definition of $\tilde{M}$ we have $T_j^x \neq \emptyset$. *)

(B) Simulate $\tilde{M}$ on input t_x. If $\tilde{M}$ on t_x requires the next input without producing a hypothesis, then set $FLAG(x) = FLAG(x-1)$, output $f(FLAG(x), x)$ and request the next input.
Else let $k = \tilde{M}(t_x)$.
(* By definition of $\tilde{M}$ we have $T_k^x \neq \emptyset$. *)
If $k = FLAG(x-1)$, then set $FLAG(x) = FLAG(x-1)$, output $f(FLAG(x), x)$ and request the next input.
If $k \neq FLAG(x-1)$, then test whether $T^x_{FLAG(x-1)} \subseteq L_{f(k,x)}$.
If $T^x_{FLAG(x-1)} \not\subseteq L_{f(k,x)}$, set $FLAG(x) = FLAG(x-1)$, output $f(FLAG(x), x)$ and request the next input.
Finally, if $T^x_{FLAG(x-1)} \subseteq L_{f(k,x)}$, then set $FLAG(x) = k$, output $f(FLAG(x), x)$ and request the next input.

It remains to show that $\mathcal{L} \subseteq ESMON(M)$.

Claim 1. M works strong–monotonically.

Since f is a strong–monotonic limiting recursive compiler from $\mathcal{G}$ into $\mathcal{L}$, the following condition is satisfied. If $L(G_j) = L$, then $L_{f(j,x)} \subseteq L$ for all $x \in \mathbb{N}$. This can be seen as follows. Let $k = lim_{x\to\infty} f(j, x)$. In particular, f is a limiting recursive compiler, and hence, $L(G_j) = L_k = L$. Moreover, f works strong–monotonically. Therefore we get

$L_{f(j,x)} \subseteq L_{f(j,x+1)} \subseteq ... \subseteq L_{lim_{x\to\infty}} f(j,x) = L_k = L = L(G_j)$.

This observation directly yields that M works strong–monotonically, if $FLAG(x) = FLAG(x-1)$.

Now, let $FLAG(x) \neq FLAG(x-1)$. If M outputs its first guess, then $T_j^x \neq \emptyset$. Due to assertion (2) of Lemma 1 we get $T_j^{x+r} \neq \emptyset$ for all $r \in \mathbb{N}$. If M, after having read t_x changes its mind, say from j to k, then we additionally know that $T_k^x \neq \emptyset$. Moreover, by assertion (2) of Lemma 1 again we obtain $T_k^{x+r} \neq \emptyset$ for all $r \in \mathbb{N}$. Hence, the condition $T^x_{FLAG(x-1)} \neq \emptyset$ is always satisfied.
Therefore, we have the following situation: $k = \tilde{M}(t_x)$, $k \neq FLAG(x-1)$ and $\emptyset \neq T^x_{FLAG(x-1)} \subseteq L_{f(k,x)} = L_{f(FLAG(x),x)}$.

By construction as well as in accordance with the definition of f and the observation made above we get:

$$L_{f(FLAG(x-1),x-1)} \subseteq L(G_{FLAG(x-1)}).$$

Hence, the assumptions of assertion (3) of Lemma 1 are fulfilled. Thus,

$T^x_{FLAG(x-1)} \subseteq L_{f(k,x)}$ implies $L(G_{FLAG(x-1)}) \subseteq L_{f(k,x)}$.

Taking these facts altogether, we obtain that $L_{f(FLAG(x-1),x-1)} \subseteq L_{f(k,x)} = L_{f(FLAG(x),x)}$. Hence, M works strong-monotonically.

Claim 2. M infers L from t.

By assumption, $z = lim_{x\to\infty}\tilde{M}(t_x)$ exists and satisfies $L = L(G_z)$. By assertion (2) of Lemma 2 there is a y_0 such that $T_z^y \neq \emptyset$ for all $y > y_0$. In particular, f is a limiting recursive compiler from $\mathcal{G}$ into $\mathcal{L}$. Hence, for all j with $L = L(G_j)$ there exists $n = lim_{x\to\infty} f(j,x)$ and it satisfies $L_n = L(G_j) = L$.

Taking the latter facts into account we see that M has to output sometimes a first hypothesis $f(j,x)$. If $j = z$, then all subsequent hypotheses have the form $f(j, x+r)$, $r \in \mathbb{N}$, and M converges to a correct hypothesis.

Now, let $j \neq z$. Then the IIM $\tilde{M}$ has not yet converged. Consequently, it suffices to show that $M(t_x) = f(z, x)$ for almost all $x \in \mathbb{N}$. If $x > y_0$, then $T_z^x \neq \emptyset$. Moreover, $L = L_{f(z,x)}$ for almost all $x \in \mathbb{N}$. As we have seen above, $L(G_{FLAG(x-1)}) \subseteq L$, and by assertion (2) of Lemma 1 we additionally know that $T^x_{FLAG(x-1)} \subseteq L$. Therefore, M sometimes verifies $T^x_{FLAG(x-1)} \subseteq L_{f(z,x)}$ and it sets $FLAG(x) = z$. Since $\tilde{M}$ cannot reject z, the IIM M cannot change its $FLAG$ in some subsequent step. Hence, M converges to a correct hypothesis for L.

q.e.d.

Theorem 5. *Let $\mathcal{L}$ be an indexed family and let $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ be a space of hypotheses such that $\mathcal{L} \in WMON$ w.r.t. $\mathcal{G}$. Then we have:*

$\mathcal{L} \in EWMON$ if and only if there is a limiting recursive weak–monotonic compiler from $\mathcal{G}$ into $\mathcal{L}$.

In particular, the latter theorems and Theorem 1 together prove the *non–existence of limiting recursive monotonic compilers* between some recursively enumerable families of uniformly recursive languages.

5. Class Comprising and Exact Learning

As we have seen in Section 3, when dealing with monotonic inference, class comprising learning is almost always more powerful than class preserving inference which itself is superior to exact learning. In particular, the results obtained give strong evidence that exclusively changing the descriptions for the objects to be learnt as well as their enumeration does not suffice to get learning algorithms of maximal power. Therefore, we are interested in knowing what kind of languages has to be supplemented to the spaces of hypotheses in order to design superior inference procedures. Moreover, we ask whether or not these added languages may be learnt themselves as well. As the following theorems show, the answer to these questions strongly depends on the type of monotonicity requirement involved. A careful analysis of our proof that $CWMON^d = LIM$ yields the first somehow unexpected result.

Theorem 6. *For all indexed families $\mathcal{L}$ we have:*

If $\mathcal{L} \in CWMON^d$, then there is a hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ comprising $\mathcal{L}$ such that $L(\mathcal{G}) \in EWMON^d$.

One possibility to prove the above theorem consists in choosing $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ as the canonical enumeration of all finite intersections of elements of $\mathcal{L}$. Hence, this construction may be performed independently of the IIMs witnessing $\mathcal{L} \in CWMON^d$.

Looking at dual strong–monotonic inference, the situation completely changes. In this setting, one is required to add grammars to the space of hypotheses that describe languages being *not learnable themselves.* This is stated by the next theorem.

Theorem 7. *Let $\mathcal{L}$ be any indexed family satisfying $\mathcal{L} \in CSMON^d \setminus SMON^d$. Then there is no space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ of hypotheses such that $\mathcal{L} \subset L(\mathcal{G})$, and $L(\mathcal{G}) \in SMON^d$.*

Proof. Suppose the converse, i.e., there is a space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ of hypotheses such that $(L(G_j))_{j\in\mathbb{N}} \in ESMON^d$ w.r.t. $(L(G_j))_{j\in\mathbb{N}}$. By assumption, $\mathcal{L} \in CSMON^d \setminus SMON^d$, and hence, $\mathcal{L} \subset \{L(G_j) \mid j \in \mathbb{N}\}$. Moreover, due to Theorem 1 we know that $FIN = ESMON^d$. Therefore, $(L(G_j))_{j\in\mathbb{N}} \in ESMON^d$ implies $(L(G_j))_{j\in\mathbb{N}} \in FIN$. Furthermore, by Theorem 1 we know that $FIN = EFIN$. Consequently, there is an IIM M that finitely infers $(L(G_j))_{j\in\mathbb{N}}$ w.r.t. $(L(G_j))_{j\in\mathbb{N}}$. On the other hand, $\mathcal{L} \subset \{L(G_j) \mid j \in \mathbb{N}\}$. Hence, M finitely infers $\mathcal{L}$, too. Applying Theorem 1 once again, we conclude $\mathcal{L} \in SMON^d$. This contradicts the assumption.
q.e.d.

The next theorem has some special features distinguishing it from the previous ones. As we have seen, in case one is dealing with dual weak–monotonic inference, there is a unique and simple way to complete the space of hypotheses appropriately. On the other hand, the requirement to work dual strong–monotonically does not allow at all a completion of the space of hypotheses such that the whole space becomes learnable in the sense of $SMON^d$. So, what can be asserted concerning weak–monotonic learning. The answer is twofold. First, we have been able to prove a theorem yielding the same statement for $CWMON$ than Theorem 6 does for $CWMON^d$. However, the construction of the desired space of hypotheses is much more complicated. Fortunately, it remains uniform in the IIMs witnessing $\mathcal{L} \in CWMON(M)$. This result has been achieved by first characterizing $CWMON$ in terms of recursive and finite tell–tales. The resulting space of hypotheses $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ is the canonical enumeration of all those languages $L(G_j)$ possessing a non–empty tell–tale. For the purpose of characterizing $CWMON$ as described it has been necessary to modify considerably the proof techniques of Lange and Zeugmann (1992).

Theorem 8. *For all indexed families $\mathcal{L}$ we have:*

If $\mathcal{L} \in CWMON$, then there is a hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ comprising $\mathcal{L}$ such that $L(\mathcal{G}) \in EWMON$.

The situation concerning $CSMON$ still remains unsolved. We shall discuss this and other open problems in the next section.

6. Conclusions and Open Problems

Learning by generalization and specialization, respectively, are modes of inference favored in various domains of machine learning. We have studied the capabilities of such learning algorithms in dependence on the choice of the space of hypotheses. The results obtained strongly recommend to carefully choose the space of hypotheses in order to obtain superior learning algorithms. Moreover, we provided additionally insight into the problem under what circumstances the choice of the space of hypotheses does considerably influence the learning power.

Furthermore, we have established conditions under what circumstances hypothesis spaces are equivalent. These results may be applied, at least conceptually, in the design of learning algorithms. Suppose, one is interested in learning an indexed family $\mathcal{L}$ strong–monotonically w.r.t. $\mathcal{L}$. Then, Theorem 4 offers the following possibility to proceed. First, one may define a *class preserving* hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ that possesses several useful properties w.r.t. the design of a learning algorithm. Applying these properties, it might be easier to define an IIM learning $\mathcal{L}$ w.r.t. $\mathcal{G}$. Second, in order to achieve the original learning goal, now it suffices to construct a strong–monotonic limiting recursive compiler from $\mathcal{G}$ into $\mathcal{L}$. The latter task might be easier than the direct solution of the learning problem. This is caused by the following observation. When one has to design a strong–monotonic limiting recursive compiler, it suffices to deal with *one text,* while the construction of a learning algorithm has to be done w.r.t. *all positive presentations.*

We view these results as a further step toward a viable theory for machine learning.

However, several problems remained open. First, it would be very interesting to know whether Theorem 8 remains valid if $CWMON$ is replaced by $CSMON$. The main difficulty one has to deal with when trying to solve this problem consists in the unpredictable behavior of IIMs on data of languages that belong to the class comprising space of hypotheses but not to the indexed family of target languages. Next, it would be desirable to extend the achieved results to more sophisticated notions of monotonicity (cf. Kapur, 1992, Lange and Zeugmann, 1992). Finally, it seems very promising to combine the approach presented with probabilistic modes of inference. In the setting of inductive inference of recursive functions Freivalds, Kinber and Wiehagen (1988) have shown that there are spaces of hypotheses $\mathcal{H}$ such that non–exactly learnable function classes might become inferable with probability 1 with respect to $\mathcal{H}$.

7. References

[1] ADLEMAN, L.M., AND BLUM,M. (1991), Inductive inference and unsolvability, *Journal of Symbolic Logic* **56**, 891 - 900.

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

[3] ANTHONY, M. AND BIGGS, N. (1992), "Computational Learning Theory," Cambridge University Press, Cambridge.

[4] FREIVALDS, R., KINBER, E.B., AND WIEHAGEN, R. (1988), Probabilistic versus deterministic inductive inference in nonstandard numberings. *Zeitschrift f. Mathematische Logik und Grundlagen der Mathematik* **34** 531 - 539.

[5] FREIVALDS, R., KINBER, E.B., AND WIEHAGEN, R. (1992), Convergently versus divergently incorrect hypotheses in inductive inference, GOSLER Report 02/92, January 1992, Fachbereich Mathematik und Informatik, TH Leipzig.

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

[7] JANTKE, K.P. (1991) Monotonic and Non–monotonic Inductive Inference, *New Generation Computing* **8**, 349 - 360.

[8] KAPUR, S. (1992), Monotonic language learning, *in* "Proceedings 3rd Workshop on Algorithmic Learning Theory," October 1992, Tokyo, JSAI, pp. 147 - 158.

[9] KAPUR, S., AND BILARDI, G. (1992), Language learning without overgeneralization, *in* "Proceedings 9th Annual Symposium on Theoretical Aspects of Computer Science," Cachan, February 13 - 15, (A. Finkel and M. Jantzen, Eds.), Lecture Notes in Computer Science Vol. 577, pp. 245 - 256, Springer-Verlag, Berlin.

[10] KINBER, E.B. (1993), personal communication.

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

[12] LANGE, S., AND ZEUGMANN, T. (1993a), Monotonic versus non–monotonic language learning, *in* "Proceedings 2nd International Workshop on Nonmonotonic and Inductive Logic," December 1991, Reinhardsbrunn, Lecture Notes in Artificial Intelligence 659, pp. 254 - 269, Springer-Verlag, Berlin.

[13] LANGE, S., AND ZEUGMANN, T. (1993b), Language learning with a bounded number of mind changes, *in* "Proceedings 10th Annual Symposium on Theoretical Aspects of Computer Science, Würzburg, Februar 1993," (P. Enjalbert, A. Finkel and K.W. Wagner, Eds.), Lecture Notes in Computer Science 665, pp. 682 - 691, Springer-Verlag, Berlin.

[14] LANGE, S., AND ZEUGMANN, T. (1993c), Learning recursive languages with bounded mind changes, *International Journal of Foundations of Computer Science*, to appear.

[15] LANGE, S., AND ZEUGMANN, T. (1993d), The learnability of recursive languages in dependence on the hypothesis space, GOSLER–Report 20/93, FB Mathematik und Informatik, TH Leipzig.

[16] LANGE, S., ZEUGMANN, T., AND KAPUR, S. (1992), Class preserving monotonic language learning, submitted to *Theoretical Computer Science*, and
GOSLER–Report 14/92, FB Mathematik und Informatik, TH Leipzig.

[17] MACHTEY, M., AND YOUNG, P. (1978) "An Introduction to the General Theory of Algorithms," North-Holland, New York.

[18] MUKOUCHI, Y. (1992a), Characterization of finite identification, *in* "Proceedings Analogical and Inductive Inference," October 1992, Dagstuhl Castle, (K.P. Jantke, Ed.), Lecture Notes in Artificial Intelligence 642, pp.260 - 267, Springer-Verlag, Berlin.

[19] MUKOUCHI, Y. (1992b), Inductive Inference with Bounded Mind Changes, *in* "Proceedings 3rd Workshop on Algorithmic Learning Theory," October 1992, Tokyo, JSAI, pp. 125 - 134.

[20] NATARAJAN, B.K. (1991), "Machine Learning, A Theoretical Approach," Morgan Kaufmann Publishers, Inc.

[21] OSHERSON, D., STOB, M., AND WEINSTEIN, S. (1986), "Systems that Learn, An Introduction to Learning Theory for Cognitive and Computer Scientists," MIT-Press, Cambridge, Massachusetts.

[22] SHINOHARA, T. (1986), Studies on inductive inference from positive data, PhD thesis, Kyushu University, Japan.

[23] SHINOHARA, T. (1990), Inductive Inference from Positive Data is Powerful, *in* "Proceedings 3rd Annual Workshop on Computational Learning Theory," (M. Fulk and J. Case, Eds.), pp. 97 - 110, Morgan Kaufmann Publishers Inc.

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