The Learnability of Recursive Languages in Dependence on the Hypothesis Space

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

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

July 28, 1993

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. Definition 1). 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 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. Ka-

pur 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 *almost always 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 is a characterization result stating that exact learning is of the same power than class preserving inference if and only if there is a limiting recursive compilers satisfying an appropriate monotonicity requirement (cf. Theorem 10. 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 11 and Corollary 14). Most of the latter results mentioned 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. We adopt the following concepts from recursion theory, (cf. Machtey and Young, 1978). By φ_1, φ_2, φ_3, ... we denote any fixed acceptable programming system for the partial recursive functions over $\mathbb{N}$. φ_k is the partial recursive function computed by program k in the φ–system. Furthermore,

let Φ_1, Φ_2, Φ_3, ... be any associated complexity measure for the φ–system. Intuitively, $\Phi_k(x)$ denotes the number of steps φ-program k runs on input x before, if ever, halting. Let $k, x \in \mathbb{N}$. If $\varphi_k(x)$ is defined (abbr. $\varphi_k(x)\downarrow$) then we also say that $\varphi_k(x)$ converges. Otherwise, $\varphi_k(x)$ is not defined or, synonymously, it diverges (abbr. $\varphi_k(x)\uparrow$).

By $\langle .,. \rangle : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ we denote Cantor's pairing function. Moreover, we use $\langle .,.,. \rangle : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ to denote the following encoding: $\langle x, y, z \rangle =_{df} \langle x, \langle y, z \rangle \rangle$ for all $x, y, z \in \mathbb{N}$.

In the sequel we assume familiarity with formal language theory (cf. e.g. Salomaa, 1973). 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 denotes the initial segment of t of length x, and $t_x^+ =_{df} \{s_k \mid k \leq x\}$.

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 follows. 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. For all $x \geq 1$ we define:

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

In the sequel we deal with the learnability of indexed families of uniformly recursive languages defined as follows (cf. Angluin, 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 $\mathcal{L}$. The meaning will be clear from the context.

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

At this point we 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. Now we define some concepts of learning. We start with learning in the limit.

Definition 1. (Gold, 1967)

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

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

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

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 from text t with respect to $\mathcal{G}$ iff it outputs only a single and correct hypothesis j, i.e., $L = L(G_j)$, and stops.

Furthermore, M FIN–identifies L, iff M FIN–identifies L ¿from every $t \in text(L)$. We set:
$FIN(M) = \{L \in \mathcal{L} \mid M \text{ FIN–identifies } L\}$ *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.

Finally we define conservative learning originally introduced by Angluin (1980).

Definition 4. (Angluin, 1980)

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

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

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

$CONSERVATIVE(M)$ as well as the collections of sets $CONSERVATIVE$ are defined in an analogous manner as above.

This is a good point to make the following convention. Let M be an IIM, let t be any text and $x \in \mathbb{N}$. If M on input t_x outputs j_x such that $t^+_x \subseteq L(G_{j_x})$, then we call j_x *consistent*. Otherwise, j_x is said to be *inconsistent*. Then Property (2) of the latter definition actually asserts that a conservatively working IIM is allowed to change its mind if and only if it detects an inconsistency with the data it has seen so far.

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 $range(\mathcal{L}) = \{L(G_j) \mid j \in \mathbb{N}\}$.

We finish this section with the following proposition from Lange and Zeugmann (1993a), where ϵ denotes the empty string.

Proposition 1. *For all* $\lambda \in \{E, \epsilon, C\}$ *we have:*

$\lambda CONSERVATIVE = \lambda WMON$.

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 the theorem $EWMON \subset ELIM$ of Angluin (1980). 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 the indexed family witnessing $ELIM \setminus EWMON \neq \emptyset$ 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 with respect to which the desired IIM $\hat{M}$ may infer U semantically finite (cf. Wiehagen, 1991). However, $\hat{M}$ does not work conservatively. Hence, when dealing with inductive inference of recursive functions, conservative learning and semantically finite inference are *different* things. On the other hand, in the setting of learning indexed families the situation considerably changes. As long as one has to learn class preservingly, conservativeness is the only way to achieve semantically finite inference (cf. Lange and Zeugmann, 1993c). However, the class comprising case remained open. As we shall see, some of our results shed additional light to this problem.

First 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, nevertheless, $CWMON \subset LIM$. In particular, the latter statement suggests that, when learning from positive data, non-semantically finite working IIMs might be 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.

Proposition 2.

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

Proof. We shall prove the Proposition by a series of theorems presented below. Note that 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). $EFIN = FIN = CFIN$ will be established in Theorem 1. Theorem 2 yields $\lambda WMON \subset \lambda WMON^d$, for any $\lambda \in \{E, \epsilon, C\}$. Next we separate exact learning and exact dual weak-monotonic inference (cf. Theorem 3). A simple modification of the latter proof additionally yields $EWMON^d \subset WMON^d$ (cf. Theorem 4). Thereafter we generalize the proof technique developed and separate $WMON^d$ and LIM (cf. Theorem 5). Furthermore, Theorem 8 relates class comprising dual strong-monotonic learning to exact weak-monotonic inference. Finally, we completely compare strong-monotonic inference and weak-monotonic learning in dependence on the allowed class of hypothesis spaces (cf. Theorem 9). Taking all these results into account, the Proposition follows.

q.e.d.

Theorem 1. $EFIN = FIN = CFIN$.

Proof. Obviously, it suffices to show that $CFIN \subseteq EFIN$. Let $\mathcal{L} \in CFIN$. Then there are a class comprising hypothesis space $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ and an IIM such that $\mathcal{L} \subseteq CFIN(M)$ w.r.t. $\mathcal{G}$. We continue with the definition of an IIM $\hat{M}$ that finitely infers $\mathcal{L}$ w.r.t. $\mathcal{L}$ itself.

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

$\hat{M}(t_x) =$ "Simulate M on t_y, for $y = 1, \ldots, x$. If M has produced a hypothesis, say on t_{y_0}, then goto (A). Otherwise request the next input.

(A) Compute the minimal index j such that $range(t_{y_0}) \subseteq L_j$. Output the hypothesis j and stop."

By assumption, M finitely infers L from t w.r.t. $\mathcal{G}$. Hence, there exists an y_0 such that M on t_{y_0} outputs for the first time a hypothesis. Moreover, this hypothesis is correct for L, i.e., we have $L = L(G_{M(t_{y_0})}) \in \mathcal{L}$. Additionally, since $\mathcal{L}$ is an indexed family, it can be effectively tested whether or not $range(t_{y_0}) \subseteq L_j$ for $j = 1, 2, \ldots$. Furthermore, since $t \in text(L)$, the

execution of (A) must eventually terminate. Hence, $\hat{M}$ is indeed an IIM. It remains to show that $L = L_j$. By construction, $range(t_{y_0}) \subseteq L_j$. Therefore, t_{y_0} is an initial segment of a text for L_j. Moreover, M when fed with t_{y_0} outputs for the first time a hypothesis. Since $L_j \in \mathcal{L}$, this hypothesis is a correct one, i.e., $L_j = L(G_{M(t_{y_0})})$. On the other hand, $L = L(G_{M(t_{y_0})})$, and thus $L = L_j$. This proves the Theorem.

q.e.d.

Theorem 2. *For any $\lambda \in \{E, \epsilon, C\}$ we have: $\lambda WMON \subset \lambda WMON^d$.*

Proof. As an immediate consequence of the corresponding definitions one gets that $\lambda CONSERVATIVE \subseteq \lambda WMON^d$. By Proposition 1 we additionally know that weak-monotonic inference equals conservative learning. Hence, it suffices to show the following lemma.

Lemma. *$EWMON^d \setminus CWMON \neq \emptyset$.*

Proof. The desired indexed family $\mathcal{L}_{ewd}$ is defined as follows. For all $k \in \mathbb{N}$ we set $L_{\langle k,1\rangle} = \{a^k b^n \mid n \in \mathbb{N}\}$. 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 $L_{\langle k,j\rangle} = L_{\langle k,1\rangle}$.

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

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

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

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

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

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

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

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

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

Since M has to infer $L_{\langle k,1\rangle}$ in particular from t, there has to be a least z such that M on input t_z computes a hypothesis j satisfying $t_z^+ \cup \{a^k b^{z+1}\} \subseteq L(G_j)$. Moreover, the test

whether or not $t_z^+ \cup \{a^k b^{z+1}\} \subseteq L(G_j)$ can be effectively performed, since membership in $L(G_j)$ is uniformly decidable. By the definition of a complexity measure, instruction (A2) is effectively executable. Hence, $\mathcal{A}$ is an algorithm.

It remains to show that $\varphi_k(k)$ diverges, if $\neg\ \Phi_k(k) \leq z$. Suppose the converse; then there exists a $y > z$ with $\Phi_k(k) = y$. In accordance with the definition of $\mathcal{L}_{ewd}$, 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, Claim B is proved.

Claim C. $\mathcal{L}_{ewd} \in EWMON^d$

We have to show that there is an IIM M inferring $\mathcal{L}_{ewd}$ dual strong-monotonically w.r.t. $\mathcal{L}_{ewd}$.

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 $\langle k, 1\rangle$ and request the next input.

(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 $\langle k, 1\rangle$ and request the next input. Otherwise, goto (B).

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

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 $L_{\langle k,1\rangle}$ 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 $L_{\langle k,j\rangle} \subset L_{\langle k,1\rangle}$. Hence, M converges to a correct hypothesis for L. This proves Claim C, and hence the lemma follows.

q.e.d.

Next to, we deal with dual weak-monotonic inference. $EWMON^d \subset ELIM$ 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 $EWMON^d \subset WMON^d$. However, we reprove this assertion here, since our new proof delivers some additional insight which we point out below.

Theorem 3. $ELIM \setminus EWMON^d \neq \emptyset$.

Proof. We prove the theorem via the following indexed family. For all $k_1, k_2 \in \mathbb{N}$ we set: $L_{\langle k_1,k_2,1\rangle} = \{a^{\langle k_1,k_2\rangle} b^z \mid z \in \mathbb{N}\}$.

For $j > 1$ we define $L_{\langle k_1,k_2,j\rangle}$ as follows:

Case 1. $\neg\Phi_{k_1}(k_1) \leq j$.

We set $L_{\langle k_1,k_2,j\rangle} = L_{\langle k_1,k_2,1\rangle}$.

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

Then we set $L_{\langle k_1,k_2,j\rangle} = \{a^{\langle k_1,k_2\rangle}b^z \mid z \leq \Phi_{k_1}(k_1)\} \cup L'_{\langle k_1,k_2\rangle}$, where $L'_{\langle k_1,k_2\rangle}$ is defined as follows via its characteristic function f.

$$f(s) = \begin{cases} 0, & \text{if } s \neq a^{\langle k_1,k_2\rangle}c^m \text{ or } s = a^{\langle k_1,k_2\rangle}c^m,\ \neg\Phi_{k_2}(k_2) = m, \\ 1, & \text{otherwise.} \end{cases}$$

After a bit of reflection it is easy to see that f is recursive. Hence, $(L_{\langle k_1,k_2,j\rangle})_{k_1,k_2,j\in\mathbb{N}}$ is an indexed family. First of all we show that $\mathcal{L}$ can be identified.

Lemma 1. $\mathcal{L} \in ELIM$.

Proof. Let $L \in \mathcal{L}$, $t \in text(L)$ and let $x \in \mathbb{N}$. We define an IIM M witnessing $\mathcal{L} \in ELIM$.

$M(t_x) =$ "If $\{a^{\langle k_1,k_2\rangle}c^m\} \subseteq t_x^+$ for some $m \in \mathbb{N}$, then goto (A). Else, execute instruction (B).

(A) Output $\langle k_1, k_2, \Phi_{k_1}(k_1)\rangle$ and request the next input.

(B) Test whether or not $\Phi_{k_1}(k_1) = x$. In case $\neg\Phi_{k_1}(k_1) = x$, output $\langle k_1, k_2, 1\rangle$ and request the next input.
Otherwise, test whether or not $t_x^+ \subseteq L_{\langle k_1,k_2,\Phi_{k_1}(k_1)\rangle}$.
In case it is, output $\langle k_1, k_2, \Phi_{k_1}(k_1)\rangle$ and request the next input. Else output $\langle k_1, k_2, 1\rangle$."

It remains to show that M infers L from t. Suppose $\{a^{\langle k_1,k_2\rangle}c^m\} \subseteq t_x^+$ for some $m \in \mathbb{N}$. In accordance with the definition of $\mathcal{L}$ we may conclude that $\Phi_{k_1}(k_1)\downarrow$. Moreover, in this case L is already uniquely determined. Hence, M converges to a correct hypothesis.

Next we suppose that no string $\{a^{\langle k_1,k_2\rangle}c^m\} \subseteq t_x^+$ does appear in t. Then we have to distinguish the following two cases.

Case 1. $\Phi_{k_1}(k_1)$ diverges.

By the definition of $\mathcal{L}$ we get that $L_{\langle k_1,k_2,j\rangle} = L_{\langle k_1,k_2,1\rangle}$ for all $j \in \mathbb{N}$. Hence, M exclusively outputs correct hypotheses.

Case 2. $\Phi_{k_1}(k_1)$ converges.

By construction, as long as M does not verify $\Phi_{k_1}(k_1) = x$, it outputs $\langle k_1, k_2, 1\rangle$. However, at this stage of the learning process it is still open whether or not $\langle k_1, k_2, 1\rangle$ is a correct guess. On the other hand, there is a x such that $\Phi_{k_1}(k_1) = x$. As long as the data M receives do not contradict the possible hypothesis $\langle k_1, k_2, \Phi_{k_1}(k_1)\rangle$, M outputs it. Note that, by definition of $\mathcal{L}$, the guess $\langle k_1, k_2, \Phi_{k_1}(k_1)\rangle$ is the minimal consistent one. Therefore, if M converges to it, it again infers L. Otherwise, the hypothesis $\langle k_1, k_2, \Phi_{k_1}(k_1)\rangle$ is provably misclassified. Then M outputs $\langle k_1, k_2, 1\rangle$ in any subsequent step. In accordance with $\mathcal{L}$'s definition, this hypothesis is again correct. This proves the lemma.

Note that the IIM defined above performs, if ever, only one mind change not caused by an inconsistency. However, this mind change may violate the dual weak-monotonicity constraint. Anyway, in order to show $\mathcal{L} \notin EWMON^d$ we need some additional insight into the behavior of *any* IIM inferring $\mathcal{L}$. The following lemma provides the desired information.

Lemma 2. *Let M be any IIM witnessing $\mathcal{L} \in ELIM$. Then we have:*
For all k_2 there are numbers $k_1, x, n \in \mathbb{N}$ such that

(1) $M(t_x) = \langle k_1, k_2, n \rangle$,

(2) $L_{\langle k_1,k_2,n \rangle} = L_{\langle k_1,k_2,1 \rangle}$,

(3) $\Phi_{k_1}(k_1) > x$ *and* $\varphi_{k_1}(k_1)\downarrow$,

where t is the lexicographically ordered text of $L_{\langle k_1,k_2,1 \rangle}$.

Proof. Suppose the converse. Then there is a k_2 such that for all k_1, x, n we have: $M(t_x) = \langle k_1, k_2, n \rangle$ and $L_{\langle k_1,k_2,n \rangle} = L_{\langle k_1,k_2,1 \rangle}$ implies $\Phi_{k_1}(k_1) \leq x$ or $\Phi_{k_1}(k_1)\uparrow$.

Assuming the latter statement we have the following claim.

Claim. Provided the latter statement is true, any program for M may be used to obtain *non-effectively* an algorithm deciding "$\varphi_{k_1}(k_1)\downarrow$."

By assumption, there is a k_2 such that for all k_1, n, x: If (1) and (2) are fulfilled, then either $\Phi_{k_1}(k_1) \leq x$ or $\Phi_{k_1}(k_1)\uparrow$. Using this k_2 we can define the following algorithm $\mathcal{A}$.

Algorithm $\mathcal{A}$: "On input k_1 execute (A1) and (A2).

(A1) Generate successively the lexicographically ordered text t of $L_{\langle k_1,k_2,1 \rangle}$ and compute $M(t_1)$, $M(t_2)$, ... until the first hypothesis $\langle k_1, k_2, n \rangle$ satisfying $L_{\langle k_1,k_2,1 \rangle} = L_{\langle k_1,k_2,n \rangle}$ is found.
Let x_0 be the least x such that $M(t_x) = \langle k_1, k_2, n \rangle$ and $L_{\langle k_1,k_2,1 \rangle} = L_{\langle k_1,k_2,n \rangle}$.

(A2) Test whether $\Phi_{k_1}(k_1) \leq x_0$.
In case it is, output "$\varphi_{k_1}(k_1)\downarrow$."
Otherwise, output "$\varphi_{k_1}(k_1)\uparrow$" and stop."

First we observe that M has to infers $L_{\langle k_1,k_2,1 \rangle}$ from its lexicographically ordered text. Hence, M should eventually output a hypothesis $\langle k_1, k_2, n \rangle$ such that $L_{\langle k_1,k_2,n \rangle} = L_{\langle k_1,k_2,1 \rangle}$. Moreover, the definition of $\mathcal{L}$ directly implies that $L_{\langle k_1,k_2,n \rangle} = L_{\langle k_1,k_2,1 \rangle}$ if and only if $\neg\Phi_{k_1}(k_1) \leq n$. Hence, the desired equality can be decided effectively. Furthermore, instruction (A2) can be accomplished effectively too. Hence, $\mathcal{A}$ is an algorithm and the execution of (A1) and (A2) must eventually terminate. Finally, by assumption we immediately obtain the correctness of $\mathcal{A}$'s output. This proves the claim.

On the other hand, the halting problem is algorithmically undecidable. Therefore, Lemma 2 follows.

Lemma 3. $\mathcal{L} \notin EWMON^d$.

Proof. Suppose the converse, i.e., $\mathcal{L} \subseteq EWMON^d(M)$ for some IIM M.

Claim 1. If there is an IIM witnessing $\mathcal{L} \in EWMON^d$, then one can effectively construct an algorithm deciding whether or not $\varphi_{k_2}(k_2)\downarrow$.

Let $K = \{k \mid \varphi_k(k)\downarrow\}$ and let $j_1, j_2, \ldots$ be any fixed effective enumeration of K. We define an algorithm $\mathcal{A}$ as follows.

Algorithm $\mathcal{A}$: "On input k_2 execute (A1) and (A2).

(A1) For $k_1 = j_1,\ j_2,\ j_3, ...$ compute successively the lexicographically ordered texts $t^1,\ t^2,\ t^3, ...$ for $L_{\langle j_1,k_2,1\rangle},\ L_{\langle j_2,k_2,1\rangle}, ...$, respectively as well as by dovetailing

$M(t^1_1),\ M(t^1_2),\ ...,$
$M(t^2_1),\ M(t^2_2),\ ...,$
–
–
–
$M(t^r_1),\ M(t^r_2),\ ...,$
–
–
–

until the first hypothesis $\langle j_z, k_2, n\rangle$ is found that satisfies

($\alpha 1$) $L_{\langle j_z,k_2,1\rangle} = L_{\langle j_z,k_2,n\rangle}$, and
($\alpha 2$) $\Phi_{j_z}(j_z) > x$, where $M(t^z_x) = \langle j_z, k_2, n\rangle$.

(* By Lemma 2, the execution of (A1) has to terminate *)

(A2) Let $s =_{df} \Phi_{j_z}(j_z)$, and $e =_{df} \langle j_z, k_2\rangle$. Furthermore, we define $\hat{t}^z_{x+r+m}$ as follows: set $r = s - x$,

$$\hat{t}^z_{x+r+m} = \underbrace{a^e b,\ ...,\ a^e b^x}_{=t^z_x},\ \underbrace{a^e b^{x+1},\ ...,\ a^e b^s}_{\text{length } r},\ \underbrace{a^e b^s,\ ...,\ a^e b^s}_{m-\text{times}}$$

For $m = 0,\ 1,\ 2,\ ...$ execute in parallel ($\beta 1$) and ($\beta 2$) until ($\beta 3$) or ($\beta 4$) happens.

($\beta 1$) Test whether $\Phi_{k_2}(k_2) = m$.
($\beta 2$) Compute $M(\hat{t}^z_{x+r+m})$.
($\beta 3$) $\Phi_{k_2}(k_2) = m$ is verified. Then output "$\varphi_{k_2}(k_2)\downarrow$."
($\beta 4$) In ($\beta 2$) a hypothesis $\langle j_z, k_2, y\rangle = M(\hat{t}^z_{x+r+m})$ is computed such that $\Phi_{j_z}(j_z) \leq y$. Then output "$\varphi_{k_2}(k_2)\uparrow$" and stop."

It remains to show that $\mathcal{A}$ behaves correctly.

Claim 2. On every input k_2, the algorithm $\mathcal{A}$ terminates.

As we have already mentioned, by Lemma 2 we know that the execution of (A1) has to terminate. Hence, it suffices to show that either ($\beta 3$) or ($\beta 4$) happens. Suppose, ($\beta 3$) does not happen. Then, for all $m \in \mathbb{N}$ we have $\neg\Phi_{k_2}(k_2) = m$. Consequently, $L'_{\langle j_z,k_2\rangle} = \emptyset$, by definition of $\mathcal{L}$. Therefore, taking into account that $\Phi_{j_z}(j_z) > x$ (cf. ($\alpha 2$)) we get that $\hat{t}^z =_{df} (\hat{t}^z_{x+r+m})_{m\in\mathbb{N}}$ is a text for $L_{\langle j_z,k_2,s\rangle}$, where $s = \Phi_{j_z}(j_z)$. Hence, M, when successively fed with $\hat{t}^z$, has to converge to a hypothesis that is correct for $L_{\langle j_z,k_2,s\rangle}$. Since M learns

exactly, this hypothesis must have the form $\langle j_z, k_2, y\rangle$. Finally, taking again $\mathcal{L}$'s definition into account we obtain $\Phi_{j_z}(j_z) \leq y$. Thus, $(\beta 4)$ must happen. This proves the claim.

Claim 3. Algorithm $\mathcal{A}$ works correctly.

Obviously, if $(\beta 3)$ happens then $\varphi_{k_2}(k_2)$ is indeed defined. Suppose, $(\beta 4)$ happens. We have to show that $\varphi_{k_2}(k_2)\uparrow$. Assume $\varphi_{k_2}(k_2)\downarrow$. Then $\Phi_{k_2}(k_2)\downarrow$, by the definition of a complexity measure. Looking at the definition of $\mathcal{L}$ we see that $L'_{\langle j_z,k_2\rangle} \neq \emptyset$. Since $M(t^z_x) = \langle j_z, k_2, n\rangle$ with $L_{\langle j_z,k_2,1\rangle} = L_{\langle j_z,k_2,n\rangle}$, every hypothesis produced by M when fed with $\hat{t}^z_{x+r+m}$ has the form $\langle j_z, k_2, y\rangle$, for some $y \in \mathbb{N}$. Moreover, since $range(\hat{t}^z_{x+r+m}) \subseteq L_{\langle j_z,k_2,n\rangle}$, the machine M cannot detect an inconsistency when, if ever, changing its mind from $\langle j_z, k_2, n\rangle$ to $\langle j_z, k_2, y\rangle$. Therefore, taking into account that M has to work dual weak-monotonically, we get $co{-}L_{\langle j_z,k_2,n\rangle} \subseteq co{-}L_{\langle j_z,k_2,y\rangle}$. The latter inclusion is equivalent to $L_{\langle j_z,k_2,n\rangle} \supseteq L_{\langle j_z,k_2,y\rangle}$. Due to the definition of $\mathcal{L}$ we conclude $L'_{\langle j_z,k_2\rangle} = \emptyset$, a contradiction. Hence $\varphi_{k_2}(k_2)\uparrow$. This proves the claim. Thus the lemma is shown and the theorem follows.

q.e.d.

The next theorem compares the power of exact and of class preserving dual weak-monotonic learning.

Theorem 4. $EWMON^d \subset WMON^d$.

Proof. In order to prove the theorem is suffices to perform a minor modification of the indexed family used in the proof of the previous theorem. This modification exclusively affects the definition of the languages $L'_{\langle k_1,k_2\rangle}$. However, for completeness we give the full definition of the used indexed family.

For all $k_1, k_2 \in \mathbb{N}$ we set:
$L_{\langle k_1,k_2,1\rangle} = \{a^{\langle k_1,k_2\rangle}b^z \mid z \in \mathbb{N}\}$.

For $j > 1$ we define $L_{\langle k_1,k_2,j\rangle}$ as follows:

Case 1. $\neg\Phi_{k_1}(k_1) \leq j$.

We set $L_{\langle k_1,k_2,j\rangle} = L_{\langle k_1,k_2,1\rangle}$.

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

Then we set $L_{\langle k_1,k_2,j\rangle} = \{a^{\langle k_1,k_2\rangle}b^z \mid z \leq \Phi_{k_1}(k_1)\} \cup L'_{\langle k_1,k_2\rangle}$, where $L'_{\langle k_1,k_2\rangle}$ is defined as follows via its characteristic function f.

$$f(s) = \begin{cases} 0, & \text{if } s \neq a^{\langle k_1,k_2\rangle}c^m \text{ or } s = a^{\langle k_1,k_2\rangle}c^m,\ \neg\Phi_{k_2}(k_2) = m \text{ or } \neg\Phi_{k_1}(k_1) \leq m, \\ 1, & \text{otherwise.} \end{cases}$$

Again, it easy to see that $(L_{\langle k_1,k_2,j\rangle})_{k_1,k_2,j\in\mathbb{N}}$ is an indexed family. Now, $\mathcal{L} \notin EWMON^d$ may be proved analogously as above.

It remains to show that $\mathcal{L} \in WMON^d$. For that purpose, first we define a class preserving hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$. This is done as follows. For all $k_1, k_2 \in \mathbb{N}$ we set $L(G_{2\langle k_1,k_2\rangle}) = L_{\langle k_1,k_2,1\rangle}$ and define $L(G_{2\langle k_1,k_2\rangle+1})$ via its characteristic function f as follows:

$$f(s) = \begin{cases} 1, & \text{if } s = a^{\langle k_1,k_2\rangle}b^z,\ \neg\Phi_{k_1}(k_1) \leq z \\ & \text{or } s = a^{\langle k_1,k_2\rangle}c^m,\ m = \Phi_{k_2}(k_2) \text{ and } \Phi_{k_1}(k_1) \leq m, \\ 0, & \text{otherwise.} \end{cases}$$

Since the predicate "$\Phi_k(x) = y$" is uniformly decidable, one gets that f is recursive. In particular, this is the point why we have modified the definition of $\mathcal{L}$. Moreover, there is standard algorithm that transforms any program of a recursive characteristic function into a grammar that generates the same language (cf. Salomaa, 1973). Furthermore, it is easy to see that the membership problem is uniformly decidable for the defined hypothesis space $\mathcal{G}$.

The main difference between the hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ and the indexed family $\mathcal{L}$ consists in the fact that we substantially changed the description of the possibly finite languages. This can be seen as follows. Given k_1, k_2, j it is easily decidable whether or not $L_{\langle k_1,k_2,j\rangle}$ is finite. On the other hand, the set of all indices that describe finite languages in $\mathcal{L}$ is recursively enumerable but not recursive. Intuitively, no IIM can always search successfully for the consistent hypothesis of minimal cardinality for the language to be learnt. In effect, the latter property is the main reason for $\mathcal{L} \notin EWMON^d$.

Looking at the new hypothesis space $\mathcal{G}$, we see that the situation considerably changes. Only the odd indices can describe a possibly finite language.

Therefore, an IIM M dual weak-monotonically inferring $\mathcal{L}$ w.r.t. $\mathcal{G}$ may be informally designed as follows. On any text it computes the corresponding numbers k_1, k_2 from the first string it receives. Then it outputs $2\langle k_1, k_2\rangle + 1$ as long as that guess is consistent with the data seen so far. In case it detects an inconsistency, it changes its mind to $2\langle k_1, k_2\rangle$ and repeats this guess in any subsequent step. Obviously, $\mathcal{L} \subseteq WMON^d(M)$ w.r.t. $\mathcal{G}$.
q.e.d.

The inclusion $WMON^d \subset LIM$ is due to E.B. Kinber (1993). However, his proof is based on a diagonalization against all IIMs and all hypothesis spaces. We reprove this result by generalizing the proof technique presented above. The new proof offers some additional insight that is discussed later.

Theorem 5. $WMON^d \subset LIM$.

Proof. Conceptually, the following proof is similar to that one of Theorem 3. Therefore, we mainly describe the modifications that have to be performed. We prove the theorem via the following indexed family. For all $k_1, k_2 \in \mathbb{N}$ we set:
$L_{\langle k_1,k_2,1\rangle} = \{a^{\langle k_1,k_2\rangle}b^z \mid z \in \mathbb{N}\}$.

For $j > 1$ we define $L_{\langle k_1,k_2,j\rangle}$ as follows:

Case 1. $\neg\Phi_{k_1}(k_1) \leq j$.

We set $L_{\langle k_1,k_2,j\rangle} = L_{\langle k_1,k_2,1\rangle}$.

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

Then we set $L_{\langle k_1,k_2,j\rangle} = \{a^{\langle k_1,k_2\rangle}b^z \mid z \leq \Phi_{k_1}(k_1)\} \cup L'_{\langle k_1,k_2,j\rangle}$, where $L'_{\langle k_1,k_2,j\rangle}$ is defined as follows via its characteristic function f.

$$f(s) = \begin{cases} 0, & \text{if } s \neq a^{\langle k_1,k_2\rangle}c^m \text{ or } s = a^{\langle k_1,k_2\rangle}c^m,\ \neg\Phi_{k_2}(k_2) = m, \\ 1, & \text{otherwise.} \end{cases}$$

Case 3. $\Phi_{k_1}(k_1) \leq j + r$, where $r \geq 1$.

Subcase 3.1. $1 \leq r < j$.

Then we set:

$L_{\langle k_1,k_2,j+r\rangle} = \{a^{\langle k_1,k_2\rangle}b^z \mid z \leq \Phi_{k_1}(k_1) - r\} \cup L'_{\langle k_1,k_2,j+r\rangle}$, where

$$L'_{\langle k_1,k_2,j+r\rangle} = \begin{cases} \emptyset, & \text{if } \Phi_{k_2}(k_2)\uparrow, \\ \{a^{\langle k_1,k_2\rangle}c^{m+r}\}, & \text{if } m = \Phi_{k_2}(k_2). \end{cases}$$

Subcase 3.2. $r \geq j$.

We set $L_{\langle k_1,k_2,j+r\rangle} = L_{\langle k_1,k_2,2j-1\rangle}$

After a bit of reflection it is easy to see that $(L_{\langle k_1,k_2,j\rangle})_{k_1,k_2,j\in\mathbb{N}}$ is an indexed family. First of all we show that $\mathcal{L}$ can be identified.

***Lemma* 1**. *$\mathcal{L} \in LIM$.*

Proof. The lemma may be proved mutatis mutandis as Lemma 1 in the demonstration of Theorem 3.

In order to show $\mathcal{L} \notin WMON^d$ we need again some additional insight into the behavior of *any* IIM inferring $\mathcal{L}$. For that purpose, Lemma 2 above is modified as follows.

***Lemma* 2**. *Let $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ be any class preserving hypothesis space for $\mathcal{L}$ and let M be any IIM witnessing $\mathcal{L} \in LIM$ w.r.t. $\mathcal{G}$. Then we have:*
For all k_2 there are numbers $k_1, x, j \in \mathbb{N}$ such that

(1) $M(t_x) = j$,

(2) $a^{\langle k_1,k_2\rangle}b^{x+1} \in L(G_j)$,

(3) $\Phi_{k_1}(k_1) > x$ *and* $\varphi_{k_1}(k_1)\downarrow$,

where t is the lexicographically ordered text of $L_{\langle k_1,k_2,1\rangle}$.

Proof. Suppose the converse. Then there is a k_2 such that for all k_1, x, j we have: $M(t_x) = j$ and $a^{\langle k_1,k_2\rangle}b^{x+1} \in L(G_j)$ implies $\Phi_{k_1}(k_1) \leq x$ or $\Phi_{k_1}(k_1)\uparrow$.

Assuming the latter statement we have the following claim.

Claim. Provided the latter statement is true, any program for M may be used to obtain *non-effectively* an algorithm deciding "$\varphi_{k_1}(k_1)\downarrow$."

By assumption, there is a k_2 such that for all k_1, x, j: If (1) and (2) are fulfilled, then either $\Phi_{k_1}(k_1) \leq x$ or $\Phi_{k_1}(k_1)\uparrow$. Using this k_2 we can define the following algorithm $\mathcal{A}$.

***Algorithm* $\mathcal{A}$:** "On input k_1 execute (A1) and (A2).

(A1) Generate successively the lexicographically ordered text t of $L_{\langle k_1,k_2,1\rangle}$ and compute $M(t_1),\ M(t_2),...$ until the first hypothesis j satisfying $a^{\langle k_1,k_2\rangle}b^{x+1} \in L(G_j)$ is found.
Let x_0 be the least x such that $M(t_x) = j$ and $a^{\langle k_1,k_2\rangle}b^{x+1} \in L(G_j)$.

(A2) Test whether $\Phi_{k_1}(k_1) \leq x_0$.
In case it is, output "$\varphi_{k_1}(k_1)\downarrow$."
Otherwise, output "$\varphi_{k_1}(k_1)\uparrow$" and stop."

First we observe that M has to infers $L_{\langle k_1,k_2,1\rangle}$ from its lexicographically ordered text. Hence, M should eventually output a hypothesis j such that $a^{\langle k_1,k_2\rangle}b^{x+1} \in L(G_j)$. Moreover, since membership in $L(G_j)$ is uniformly decidable for all $j \in \mathbb{N}$ and all $s \in \{a,b,c\}^*$, the described test can be performed effectively. Furthermore, instruction (A2) can be accomplished effectively too. Hence, $\mathcal{A}$ is an algorithm and the execution of (A1) and (A2) must eventually terminate. Finally, by assumption we immediately obtain the correctness of $\mathcal{A}$'s output. This proves the claim.

On the other hand, the halting problem is algorithmically undecidable. Therefore, Lemma 2 follows.

Lemma 3. $\mathcal{L} \notin WMON^d$.

Proof. Suppose the converse, i.e., there is a hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ and an IIM M such that $\mathcal{L} \subseteq WMON^d(M)$ w.r.t. $\mathcal{G}$. Moreover, without loss of generality we may assume that M produces exclusively consistent hypotheses (cf. Lange et al., 1992).

Claim 1. Given $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ and any programm for M witnessing $\mathcal{L} \in WMON^d$, one can effectively construct an algorithm deciding whether or not $\varphi_{k_2}(k_2)\downarrow$.

Let $K = \{k \mid \varphi_k(k)\downarrow\}$ and let $j_1, j_2, ...$ be any fixed effective enumeration of K. We define an algorithm $\mathcal{A}$ as follows.

Algorithm $\mathcal{A}$: "On input k_2 execute (A1) and (A2).

(A1) For $k_1 = j_1,\ j_2,\ j_3,...$ compute successively the lexicographically ordered texts $t^1,\ t^2,\ t^3,...$ for $L_{\langle j_1,k_2,1\rangle},\ L_{\langle j_2,k_2,1\rangle},...$, respectively as well as by dovetailing

$M(t^1_1),\ M(t^1_2),\ ...,$
$M(t^2_1),\ M(t^2_2),\ ...,$
–
–
–
$M(t^r_1),\ M(t^r_2),\ ...,$
–
–
–

until the first hypothesis $j = j(j_z, k_2)$ is found that satisfies

($\alpha 1$) $M(t^z_x) = j$,
($\alpha 2$) $range(t^z_x) \cup \{a^{\langle j_z, k_2\rangle} b^{x+1}\} \subseteq L(G_j)$, and
($\alpha 3$) $\Phi_{j_z}(j_z) > x$.

(* By Lemma 2, the execution of (A1) has to terminate *)

(A2) Let $e =_{df} \langle j_z, k_2\rangle$. Furthermore, we define $\hat{t}^z_{x+m}$ as follows:
$\hat{t}^z_{x+m} = \underbrace{a^e b,\ ...,\ a^e b^x}_{=t^z_x},\ \underbrace{a^e b^x,\ ...,\ a^e b^x}_{m-\text{times}}$

For $m = 0,\ 1,\ 2,\ ...$ execute in parallel ($\beta 1$) and ($\beta 2$) until ($\beta 3$) or ($\beta 4$) happens.

($\beta 1$) Test whether $\Phi_{k_2}(k_2) = m$.
($\beta 2$) Compute $j_{x+m} = M(\hat{t}^z_{x+m})$.
($\beta 3$) $\Phi_{k_2}(k_2) = m$ is verified. Then output "$\varphi_{k_2}(k_2)\downarrow$."
($\beta 4$) In ($\beta 2$) a hypothesis $j_{x+m} = M(\hat{t}^z_{x+m})$ is computed such that $a^e b^{x+1} \notin L(G_{j_{x+m}})$. Then output "$\varphi_{k_2}(k_2)\uparrow$" and stop."

It remains to show that $\mathcal{A}$ behaves correctly.

Claim 2. On every input k_2, the algorithm $\mathcal{A}$ terminates.

As we have already mentioned, by Lemma 2 we know that the execution of (A1) has to terminate. Hence, it suffices to show that either ($\beta 3$) or ($\beta 4$) happens. Suppose, ($\beta 3$) does not happen. Then, for all $m \in \mathbb{N}$ we have $\neg\Phi_{k_2}(k_2) = m$. Consequently, $\Phi_{k_2}(k_2)\uparrow$. Therefore, $L'_{\langle j_z, k_2, y\rangle} = \emptyset$ for all $y \in \mathbb{N}$ by definition of $\mathcal{L}$. Due to ($\alpha 3$) we know that $\Phi_{j_z}(j_z) > x$, and thus $\hat{t}^z =_{df} (\hat{t}^z_{x+m})_{m\in\mathbb{N}}$ is a text for $L_{\langle j_z, k_2, 2s-x+1\rangle}$, where $s = \Phi_{j_z}(j_z)$. Hence, M, when successively fed with $\hat{t}^z$, has eventually to output a hypothesis j_{x+m} such that $a^e b^{x+1} \notin L(G_{j_{x+m}})$. Thus, ($\beta 4$) must happen. This proves the claim.

Claim 3. Algorithm $\mathcal{A}$ works correctly.

Obviously, if ($\beta 3$) happens then $\varphi_{k_2}(k_2)$ is indeed defined. Suppose, ($\beta 4$) happens. We have to show that $\varphi_{k_2}(k_2)\uparrow$. Assume $\varphi_{k_2}(k_2)\downarrow$. Then $\Phi_{k_2}(k_2)\downarrow$, by the definition of a complexity measure. We distinguish the following cases.

Case 1. The hypothesis $j = M(t_x)$ satisfies $L_{\langle j_z, k_2, 1\rangle} = L(G_j)$.

Then the claim follows exactly as in the proof of Theorem 3.

Case 2. The hypothesis $j = M(t_x)$ does not satisfy $L_{\langle j_z, k_2, 1\rangle} = L(G_j)$.

Since M outputs exclusively consistent guesses, we may conclude that $L(G_j)$ is finite (cf. the definition of $\mathcal{L}$). By ($\alpha 2$) we know that $a^e b^{x+1} \in L(G_j)$. On the other hand, $a^e b^{x+1} \notin L(G_{j_{x+m}})$ since ($\beta 4$) has happened. Therefore, $L(G_{j_{x+m}})$ is finite, too. Taking again the definition of $\mathcal{L}$ into account we additionally get $L(G_j)\ \#\ L(G_{j_{x+m}})$. Finally, since $range(\hat{t}^z_{x+m}) \subseteq L(G_j)$, at least one mind change in the subsequence $j,\ ...\ j_{x+m}$ of M's outputs cannot be caused by inconsistency. Hence, this mind change must satisfy the dual weak-monotonicity constraint. However, on the one hand this is impossible since

all finite languages that comprise at least t_x are incomparable, while on the other hand $range(\hat{t}^z_{x+m}) = t^+_x$ does not allow M to output a hypothesis generating an infinite language.

This proves the claim. Thus the lemma is shown and the theorem follows.
q.e.d.

We finish the study of dual weak-monotonic learning with a theorem that establishes a sufficient condition for inferable indexed families to be dual weak-monotonically learnable. For that purpose we need the following convention.

Let $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ be a hypothesis space. We say that inclusion in $\mathcal{G}$ is partially decidable iff there is a partial recursive predicate p such that for all $i, j \in \mathbb{N}$

$$p(i,j) = \begin{cases} 1, & \text{if } L(G_j) \subseteq L(G_i), \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

Now we are ready to present the announced theorem.

Theorem 6. *Let $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ be a hypothesis space and let $\mathcal{L}$ be an indexed family that can be LIM-identified w.r.t. $\mathcal{G}$. Then we have:*

If inclusion in $\mathcal{G}$ is partially decidable then there is an IIM M that dual weak-monotonically infers $\mathcal{L}$ w.r.t. $\mathcal{G}$.

Proof. By assumption, there is an IIM $\hat{M}$ such that $\mathcal{L} \subseteq LIM(\hat{M})$ w.r.t. $\mathcal{G}$. Without loss of generality we may assume that $\hat{M}$, if it has output for the first time a guess, then it outputs in any subsequent step a hypothesis. We have to define an IIM M that learns $\mathcal{L}$ dual weak-monotonically w.r.t. $\mathcal{G}$. Let $L \in \mathcal{L}$, $t \in text(L)$, and let $x \in \mathbb{N}$. We define:

$M(t_x) =$ "If $\hat{M}$, when successively fed with t_x, does not output any hypothesis, then output nothing and request the next input.
Otherwise, goto (A).

(A) If $\hat{M}$ on t_x outputs a hypothesis, say j_x, and M, when successively fed with t_{x-1} does not produce a guess, then output j_x and request the next input.
Otherwise, goto (B).

(B) Let $i_{x-1} = M(t_{x-1})$. Compute $j_x = \hat{M}(t_x)$. Furthermore, simulate the computation of $p(i_{x-1}, j_x)$ at most x steps. If this computation yields $p(i_{x-1}, j_x) = 1$, then output j_x and request the next input.
Otherwise, if $t^+_x \subseteq L(G_{i_{x-1}})$, then output i_{x-1} and request the next input. Else output j_x and request the next input."

It is easy to see that M dual weak-monotonically infers $\mathcal{L}$ w.r.t. $\mathcal{G}$. We omit the details.
q.e.d.

On the other hand, the condition stated in the latter theorem is not necessary. This can be shown as follows. Let $\mathcal{L}$ be the indexed family and $\mathcal{G}$ be the hypothesis space used in the proof of Theorem 4. Assuming inclusion in $\mathcal{G}$ is partially decidable, one can prove the halting problem to be algorithmically solvable. We leave it to the reader to elaborate this reduction.

There is one more point we want to discuss. At first glance it might seem that partial decidability of inclusion in a hypothesis space $\mathcal{G}$ should imply strong-monotonic or at least weak-monotonic learnability of any indexed family $\mathcal{L}$ that is LIM-inferable w.r.t. $\mathcal{G}$. However, this is not the case.

Theorem 7. *There is an indexed family $\mathcal{L}$ such that*

(1) $\mathcal{L} \in EWMON^d$,

(2) *inclusion in $\mathcal{L}$ is decidable,*

(3) $\mathcal{L} \notin CWMON$.

Proof. Let $\mathcal{L}_{ewd}$ be the indexed family defined in the proof of Theorem 2. Hence, (1) and (3) are fulfilled. It remains to show that inclusion in $\mathcal{L}$ is decidable. We define a predicate p as follows:

"On input $\langle k_1, y\rangle$ and $\langle k_2, n\rangle$ do the following:

(1) If $k_1 \neq k_2$, output 0.

(2) If $k_1 = k_2$, test whether or not $\neg\Phi_{k_1}(k_1) \leq max\{y, n\}$.
In case it is, output 1.
(* By construction, $L_{\langle k_1,y\rangle} = L_{\langle k_2,n\rangle}$ *)
Otherwise, if $y \leq n$ or $min\{y, n\} \geq 2\Phi_{k_1}(k_1)$, then output 1, else 0.

By definition of $\mathcal{L}$ it is easy to verify that

$$p(\langle k_1, y\rangle, \langle k_2, n\rangle) = \begin{cases} 1, \text{ if } L_{\langle k_1,y\rangle} \subseteq L_{\langle k_2,n\rangle}, \\ 0, \text{ otherwise.} \end{cases}$$

q.e.d.

Now, let us continue with the main topic of that section, i.e., establishing separations. 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 8. $CSMON^d \subset EWMON$

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 9.

(1) $WMON \quad \# \quad CSMON$

(2) $EWMON \quad \# \quad CSMON$

(3) $EWMON \quad \# \quad SMON$

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

Proof. We start with assertion (4). Within the proof of Theorem 8 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 $\mathcal{L}_{csd} \notin CSMON$.

***Lemma* 1**. $CSMON^d \setminus CSMON \neq \emptyset$.

Proof. The desired indexed family is defined as follows. For all $k \in \mathbb{N}$ we set $L_{\langle k,1 \rangle} = \{a^k b^n \mid n \in \mathbb{N}\}$. 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: $L_{\langle k,j \rangle} = L_{\langle k,1 \rangle}$

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

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

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

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

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

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

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

Let M be any IIM which strong-monotonically infers $\mathcal{L}$ w.r.t. a space of hypotheses $\mathcal{G}$ with $range(\mathcal{L}_{csd}) \subseteq L(\mathcal{G})$. First of all, we define an algorithm $\mathcal{A}$ solving the halting problem. On input $k \in \mathbb{N}$ the algorithm $\mathcal{A}$ executes the following instructions:

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

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

Since M in particular infers $L_{\langle k,1\rangle}$ on t, there has to be a least z such that M on input t_z computes a hypothesis j satisfying $t_z^+ \cup \{a^k b^{z+1}\} \subseteq L(G_j)$. Since membership in $\mathcal{G}$ is uniformly decidable, this test can be effectively performed. Moreover, $\mathcal{A}$ is an algorithm, since the predicate "$\Phi_i(y) = z$" is uniformly recursive in i, y, and z, and, therefore, instruction (A2) is effectively executable.

It remains to show that $\varphi_k(k)$ is undefined, if $\neg\ \Phi_k(k) \leq z$. Suppose the converse, i.e., $\varphi_k(k)$ is defined. Then, $\Phi_k(k) = y$ for some $y > z$. Let m be the maximal index such that $a^k b^m \in t_z^+$. In accordance with the definition of $\mathcal{L}_{csd}$, there has to be an $L \in \mathcal{L}_{csd}$, namely $L = t_z^+ \cup \{b^k a^{2y-m}\}$, such that $L \neq L(G_j)$ and $t_z^+ \subseteq L$. It is easy to verify, that $\neg\ L(G_j) \subset L$. Since t_z is an initial segment of a text $\hat{t}$ for L, M does not work strong-monotonically when inferring L on $\hat{t}$. This contradicts our assumption that $\mathcal{L}_{csd} \subseteq CSMON(M)$. Hence, the proof of Claim B is completed.

By contraposition of Claim B we obtain Claim A, since the halting problem is not recursive.

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

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 $L(G_{\langle k,1\rangle}) = \bigcup_{j\in\mathbb{N}} L_{\langle k,j\rangle}$ and $L(G_{\langle k,j+1\rangle}) = L_{\langle k,j\rangle}$.

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 $\mathcal{L}_{csd}$. It remains to show that $\mathcal{L}_{csd} \in CSMON^d$ w.r.t. $\mathcal{G}$.

The IIM M defined below dual strong-monotonically infers $\mathcal{L}_{csd}$ w.r.t. $\mathcal{G}$. Let $L \in \mathcal{L}_{csd}$, let t be any text for L, and let $x \in \mathbb{N}$.

$M(t_x) =$ "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 $\langle k, 1 \rangle$ and request the next input.

(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 $\langle k, 2 \rangle$ and request the next input. Otherwise, goto (B).

(B) Test whether or not $b^k a^m \in t_x^+$ for some $m \in \mathbb{N}$. In case it is, output $\langle k, m+1 \rangle$ and request the next input.
Otherwise, output nothing and request the next input."

Due to the definition above, M performs at most one mind change when inferring L on text t. Due to the definition of $L(G_{\langle k,1 \rangle})$, we have $L(G_{\langle k,1 \rangle}) \supseteq L_{\langle k,j \rangle}$, for all $j \in \mathbb{N}$. This guarantees that this mind change fulfils the monotonicity requirement. Finally, it is easy to verify that M converges to a correct hypothesis for L.

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).

Lemma 2. $SMON \setminus EWMON \neq \emptyset$.

Proof. The desired indexed family $\mathcal{L}_{sm} = (L_{\langle k,j \rangle})_{j,k \in \mathbb{N}}$ is defined as follows. For all $k \in \mathbb{N}$, we set $L_{\langle k,1 \rangle} = \{a^k b^n \mid n \in \mathbb{N}\}$. 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: $L_{\langle k,j \rangle} = L_{\langle k,1 \rangle}$.

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

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

Claim A. $\mathcal{L}_{sm} \notin EWMON$.

Again, we reduce the halting problem to the considered learning problem.

Claim B. If there exists an IIM M with $\mathcal{L}_{cs} \subseteq EWMON(M)$, then one can effectively construct a total recursive predicate ψ that decides whether or not $\varphi_k(k)\downarrow$.

Let M be anyIIM which weak-monotonically infers $\mathcal{L}$ w.r.t. $\mathcal{L}_{sm}$. We define the desired predicate ψ as follows. Let $k \in \mathbb{N}$, and let t be the canonical text for $L_{\langle k,1 \rangle}$.

$$\psi(k) = \begin{cases} 1, & if\ z = \mu n\{n \mid M(t_n) = \langle k,j \rangle\} \wedge \Phi_k(k) \leq max\{j,z\}, \\ 0, & if\ z = \mu n\{n \mid M(t_n) = \langle k,j \rangle\} \wedge \neg\, \Phi_k(k) \leq max\{j,z\}. \end{cases}$$

Since M in particular infers $L_{\langle k,1\rangle}$ on t, there is a z such that M on input t_z computes a hypothesis $\langle k,j\rangle$ for some $j \in \mathbb{N}$. Hence, ψ is recursive, since the predicate "$\Phi_i(y) = z$" is uniformly recursive in i, y, and z.

It remains to show that $\varphi_k(k)$ is undefined, if $\psi(k) = 0$. Suppose the converse, i.e., $\varphi_k(k)$ converges. Hence, there is a $y \in \mathbb{N}$ such that $\Phi_k(k) = y$. Furthermore, let z be the minimal index such that $M(t_z) = \langle k,j\rangle$ for some $j \in \mathbb{N}$. Due to the definition of ψ, we obtain $y > max\{j,z\}$. Moreover , $range(t_z) \subseteq L_{\langle k,y\rangle}$ as well as $L_{\langle k,y\rangle} \subset L_{\langle k,j\rangle} = L_{\langle k,1\rangle}$. Obviously, t_z is an initial segment of a text $\hat{t}$ for $L_{\langle k,y\rangle}$. Hence, M has produced an overgeneralized hypothesis when inferring $L_{\langle k,y\rangle}$ on $\hat{t}$. Thus, M does not work weak-monotonically which contradicts our assumption that $\mathcal{L} \subseteq EWMON(M)$. Therefore, $\varphi_k(k)\uparrow$, if $\psi(k) = 0$. This proves Claim B, and hence, Claim A is shown.

Claim C. $\mathcal{L}_{sm} \in SMON$.

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

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

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

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

$M(t_x) =$ "Determine the unique k such that $t_1 = a^k b^m$ for some $m \in \mathbb{N}$. Test whether or not $t_x^+ \in L(G_{2k})$. In case it is, output $2k$. Otherwise, output $2k-1$."

Obviously, M changes its mind at most once. Since $L(G_{2k}) \subseteq L(G_{2k-1})$, this mind change satisfies the monotonicity requirement. Furthermore, M converges to a correct hypothesis for L. This proves Claim C, and therefore Lemma 2 follows.

Moreover, $EWMON \setminus SMON \neq \emptyset$ is an immediate consequence of Lemma 1 and Theorem 8. Therefore assertion (3) is proved.

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 $\mathcal{L}_{csm} = (L_{\langle k,j\rangle})_{j,k\in\mathbb{N}}$. For all $k \in \mathbb{N}$ we set $L_{\langle k,1\rangle} = \{a^k b^n \mid n \in \mathbb{N}\}$. For all $k \in \mathbb{N}$ and all $j > 1$ we distinguish the following cases:

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

We set: $L_{\langle k,j\rangle} = L_{\langle k,1\rangle}$

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

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

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

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

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 almost always effectively reduced the halting problem to several monotonic learning problems. These reductions imply that the considered learning problems are at least as hard as 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 one of the theorems obtained, since it already does 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 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 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 we introduce limiting recursive compilers fulfilling a certain monotonicity demand.

Definition 6. *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*

strong-monotonic

iff

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

Now we are ready to present the announced characterization comparing the power of exact and class preserving learning algorithms under the constraint to learn strong-monotonically.

Theorem 10. *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 strong-monotonically inferring $\mathcal{L}$ 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 a 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 $(T_j^y)_{j,y\in\mathbb{N}}$ 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 $T_j^y \neq \emptyset$ for almost all $y \in \mathbb{N}$,*

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

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

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

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

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 $(T_j^y)_{j,y\in\mathbb{N}}$ 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 $T_z^y \neq \emptyset$ 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_j^x and test for all non-empty T_j^x whether or not $T_j^x \subseteq t_x^+ \subseteq L(G_j)$. 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 $T_z^y \neq \emptyset$ 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 $T_k^0 = ... = T_k^{y_0} = \emptyset$ and $\emptyset \neq T_k^{y_0+1} = T_k^{y_0+r}$ for all $r \in \mathbb{N}$.

The latter observation also holds for all j for which there is a y such that $T_j^y \neq \emptyset$. 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,

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

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.

In particular, the latter theorem and Proposition 2 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 11. *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 $L(\mathcal{G}) \in EWMON^d$.

Proof. 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 $\mathcal{L}$, (cf. Lange, Zeugmann and Kapur, 1992). Hence, this construction may be performed independently of the IIMs witnessing $\mathcal{L} \in CWMON^d$.

In the sequel, we present another proof that might be interesting in its own right. Let $\mathcal{L} \in CWMON^d$. Since $CWMON^d = ELIM$ (cf. Proposition 2), there is an IIM M which infers $\mathcal{L}$ w.r.t. the hypothesis space $\mathcal{L}$, i.e., $\mathcal{L} \subseteq ELIM(M)$. We proceed as follows. First, we define an indexed family $\hat{\mathcal{L}}$ comprising $\mathcal{L}$ and an IIM $\tilde{M}$ witnessing $\hat{\mathcal{L}} \in ELIM$. Second, on the basis of $\tilde{M}$ we construct an IIM $\hat{M}$ which dual weak-monotonically infers $\hat{\mathcal{L}}$ w.r.t. $\hat{\mathcal{L}}$. This proves the above theorem, if we choose $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ such that $(L(G_j))_{j \in \mathbb{N}} = (\hat{L}_j)_{j \in \mathbb{N}}$.

First step: For any $j \in \mathbb{N}$, we denote by L_j^{single} the unique singleton language containing the first string of the canonical text of L_j. We define the desired indexed family $\hat{\mathcal{L}} = (\hat{L}_j)_{j \in \mathbb{N}}$ as follows. For all $j \in \mathbb{N}$, we set $\hat{L}_{2j-1} = L_j$ and $\hat{L}_{2j} = L_j^{single}$.

We continue in defining the IIM $\tilde{M}$. Let $\hat{L} \in \hat{\mathcal{L}}$, $t \in text(\hat{L})$, and let $x \in \mathbb{N}$. Recall that $\mathcal{L} \subseteq ELIM(M)$.

$\tilde{M}(t_x) =$ "As long as $card(t_x^+) = 1$ goto (A). Otherwise, goto (B).

(A) Test whether or not $t_x^+ = \hat{L}_{2x}$. In case it is, output $2x$. Otherwise, output 2.

(B) If M outputs a hypothesis j on input t_x, then output $2j - 1$. Otherwise, compute $\tilde{M}(t_{x-1})$ and output it."

Obviously, $\tilde{M}$ infers $\hat{\mathcal{L}}$ w.r.t. the hypothesis space $\hat{\mathcal{L}}$. Furthermore, $\tilde{M}$ outputs in every step a hypothesis.

Second step: Now we are ready to define the desired IIM $\hat{M}$ witnessing $\hat{\mathcal{L}} \in EWMON^d$. Let $\hat{L}$ be any language in $\hat{\mathcal{L}}$, $t \in text(\hat{L})$, and $x \in \mathbb{N}$.

$\hat{M}(t_x) =$ "As long as $card(t_x^+) = 1$ goto (A). Otherwise, goto (B).

(A) Output $\tilde{M}(t_x)$ and set $FLAG(x) = 0$.

(B) If $FLAG(x-1) = 0$, then goto (b1). Otherwise, execute (b2).

(b1) Output $\tilde{M}(t_x)$. In case $\tilde{M}(t_x)$ is even set $FLAG(x) = 0$. If $\tilde{M}(t_x)$ is odd, then set $FLAG(x) = 1$.

(b2) If $\hat{M}(t_{x-1}) = \tilde{M}(t_x)$, then output $\hat{M}(t_{x-1})$. Furthermore, set $FLAG(x) = FLAG(x-1)$.
If $\hat{M}(t_{x-1}) \neq \tilde{M}(t_x)$, output $\hat{M}(t_{x-1}) - 1$ and set $FLAG(x) = 0$."

Since $\tilde{M}$ identifies $\hat{L}$ from t, there has to be a least x such that $\tilde{M}(t_x) = \tilde{M}(t_{x+r}) = j$ for all $r \geq 1$. Furthermore, $\hat{L}_j = L$. Due to our construction, we obtain $\hat{M}(t_x) = \hat{M}(t_{x+r}) = j$ for all $r \geq 1$, provided $FLAG(x-1) = 0$. Now suppose, $FLAG(x-1) = 1$. Then we have $\hat{M}(t_{x+1}) = \hat{M}(t_{x+1+r}) = j$ for all $r \geq 1$. Hence, $\hat{M}$ converges to the same correct hypothesis as $\tilde{M}$ does when fed with text t.

It remains to show that $\hat{M}$ works dual weak-monotonically. As long as $\hat{M}$ stays in stage (A), $\hat{M}$ performs at least one mind change which is caused by an inconsistency. Therefore, $\hat{M}$ works dual weak-monotonically within stage (A). If $\hat{M}$ leaves stage (A), it has detected an inconsistency, since any hypothesis output in stage (A) describes a singleton language. Therefore, any mind change is admissible when leaving stage (A). Finally, we consider the case that $\hat{M}$ is in stage (B). Let $\hat{M}(t_x) \neq \hat{M}(t_{x+1})$, for some $x \in \mathbb{N}$. We distinguish the following two cases.

Case 1: $FLAG(x) = 0$
Then, $\hat{M}(t_x)$ denotes a singleton language. Since $\hat{M}$ is in stage (B), an inconsistency is detected. Consequently, any mind change is admissible.

Case 2: $FLAG(x) = 1$
Due to our construction of $\tilde{M}$ and $\hat{M}$, the hypothesis $\hat{M}(t_x)$ is odd. Furthermore, $\hat{M}(t_{x+1}) = \hat{M}(t_x) - 1$. Since $\hat{L}_{\hat{M}(t_{x+1})} \subseteq \hat{L}_{\hat{M}(t_x)}$, $\hat{M}$ maintains the monotonicity requirement.
q.e.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 12. *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 result 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. We start with the characterization of $CWMON$.

Theorem 13. *Let $\mathcal{L}$ be an indexed family of recursive languages. Then: $\mathcal{L} \in CWMON$ if and only if there is a space of hypothesis space $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ and a recursively generable family $(\hat{T}_j)_{j\in\mathbb{N}}$ of finite non-empty sets such that*

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

(2) *for all $j \in \mathbb{N}$, $\hat{T}_j \subseteq L(\hat{G}_j)$,*

(3) *for all j, $z \in N$, if $\hat{T}_j \subseteq L(\hat{G}_z)$, then $L(\hat{G}_z) \not\subset L(\hat{G}_j)$.*

Proof. Necessity: Let $\mathcal{L} \in CWMON$. By Proposition 1, $CWMON = CCONSERV-ATIVE$. Hence, there are a hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ and an IIM M such that $\mathcal{L} \subseteq CCONSERVATIVE(M)$ w.r.t. $\mathcal{G}$. Moreover, without loss of generality we may assume that M works consistently too, i.e., whenever M on any t_x produces a hypothesis j_x, then $t_x^+ \subseteq L(G_{j_x})$ (cf. Lange et al., 1992). We proceed by showing how to construct $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$. This is done in two steps. First, we construct a space of hypotheses $\tilde{\mathcal{G}} = (\tilde{G}_j)_{j\in\mathbb{N}}$ as well as a family of finite sets $(\tilde{T}_j)_{j\in\mathbb{N}}$. Then we describe a procedure enumerating a certain subset of $\tilde{\mathcal{G}}$ which we call $\hat{\mathcal{G}}$.

For all $k, x \in \mathbb{N}$ we set $\tilde{G}_{\langle k,x\rangle} = G_k$. Furthermore, for $k, r \in \mathbb{N}$ we use t^k and τ^r to denote the canonical text of $L(G_k)$ and L_r, respectively. Then we define $(\tilde{T}_j)_{j\in\mathbb{N}}$ via the following procedure. On input j do the following:

(i) Compute $k, x \in \mathbb{N}$ with $j = \langle k, x\rangle$

(ii) Test whether there is an index $r \leq x$ such that $t_x^k = \tau_x^r$. In case there is no $r \leq x$ that does fulfil the test define $\tilde{T}_{\langle k,x\rangle} = \emptyset$, and stop. Otherwise, execute instruction (iii).

(iii) Set

$$\tilde{T}_{\langle k,x\rangle} = \begin{cases} range(t_y^k), & \text{if} \quad y = min\{z \mid z \leq x,\ M(t_z^k) = k\}, \\ \emptyset, & \text{otherwise.} \end{cases}$$

Obviously, instruction (i) can be executed effectively. The test in instruction (ii) can be accomplished algorithmically, since $\mathcal{L}$ is an indexed family. Finally, in accordance with the definition of an IIM, it is effectively decidable whether or not $M(t_z^k) = k$. Hence, the family $(\tilde{T}_{\langle k,x\rangle})_{k,x\in\mathbb{N}}$ is recursively generable and all sets are finite. The desired space

of hypotheses $\hat{\mathcal{G}}$ is obtained from $\tilde{\mathcal{G}}$ by simply striking off all grammars $\tilde{G}_{\langle k,x\rangle}$ for which $\tilde{T}_{\langle k,x\rangle} = \emptyset$. Analogously, $(\hat{T}_j)_{j\in\mathbb{N}}$ is obtained from $(\tilde{T}_j)_{j\in\mathbb{N}}$. Obviously, $(\hat{T}_j)_{j\in\mathbb{N}}$ is a recursively generable family of finite, non-empty sets. For the sake of notational convenience we refer to $\hat{T}_j$ as to $\hat{T}_{\langle k,x\rangle}$, i.e., we omit the corresponding mapping yielding the enumeration of the sets $\hat{T}_j$ from that one for the sets $\tilde{T}_z$.

It remains to show that $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ and $(\hat{T}_j)_{j\in\mathbb{N}}$ do fulfil the announced properties. By construction, Property (2) is obviously satisfied. For the purpose of proving (1), let $L \in \mathcal{L}$. We have to show that L possesses at least one description in $\hat{\mathcal{G}}$, i.e., an index j such that $L = L(\hat{G}_j)$. In accordance with our construction it suffices to prove that there are k, x satisfying $L = L(\tilde{G}_{\langle k,x\rangle})$ and $\tilde{T}_{\langle k,x\rangle} \neq \emptyset$. Let t^L be L's canonical text. In particular, M has to infer L from t^L. Hence, there are k, y_0 such that for all $z < y_0$, $M(t^L_z) \neq k$, $M(t^L_{y_0}) = k$ and $L = L(G_k)$. Moreover, since $L \in \mathcal{L}$ there is a least index r_0 such that $L = L_{r_0}$. Let $x_0 = max\{y_0, r_0\}$. Then $\tilde{T}_{\langle k,x_0\rangle} = range(t^L_{y_0})$, and $L = L(\tilde{G}_{\langle k,x_0\rangle})$.

Finally, we prove (3). Suppose the converse, i.e., there are indices j, z such that $\hat{T}_j \subseteq L(\hat{G}_z)$ and $L(\hat{G}_z) \subset L(\hat{G}_j)$. Let k, m, x, $y \in \mathbb{N}$ be the numbers satisfying $j = \langle k, x\rangle$ and $z = \langle m, y\rangle$. Moreover, let $s_1, ..., s_n$ be the strings of $\hat{T}_j$ written as the initial segment of the canonical text of $L(\hat{G}_j)$ up to the minimal length n such that $M(s_1, ..., s_n) = k$ for the first time. By construction, $L(\hat{G}_j) = L(G_k)$. The difficulty we have to deal with is that we cannot conclude $L(G_k) \in \mathcal{L}$. If $L(G_k) \notin \mathcal{L}$, then nothing is known a priorily concerning M's behavior on initial segments of texts for $L(G_k)$. With the next claim we provide the information needed in order to finish the proof.

Claim A. For all $u \leq n$ we have $M(s_1, ..., s_u) \neq m$.

By assumption, $\hat{T}_j \subseteq L(\hat{G}_z)$ and $L(\hat{G}_z) = L(G_m)$. Consequently, for all $u \leq n$ the sequence $s_1, ..., s_u$ is an initial segment of *some* text for $L(\hat{G}_z)$. Suppose, $M(s_1, ..., s_u) = m$. Then it should hold that $\{s_1, ..., s_n\} \not\subseteq \hat{T}_j$, contradicting $\hat{T}_j = \{s_1, ..., s_n\}$. This can be seen as follows. Remembering the construction of the sets $\hat{T}_j$. Since $\hat{T}_j \neq \emptyset$, the finite sequence $s_1, ..., s_n$ is an *initial segment* of the canonical text of some language $L \in \mathcal{L}$, too. Consequently, the IIM M, when successively fed with $s_1, ..., s_n$ has to work conservatively and consistently. Therefore, if M changes its mind ¿from m to k (not necessarily in one step) when successively receiving $s_1, ..., s_n$ then every mind change has to be caused by a detected inconsistency with the text seen so far. Hence, if $m = M(s_1, ..., s_u)$, $k = M(s_1, ..., s_n)$ then, by consistency, we know that $\{s_1, ..., s_u\} \subseteq L(G_m)$ as well as $\{s_1, ..., s_n\} \subseteq L(G_k)$. Moreover, $\{s_1, ..., s_n\} \not\subseteq L(G_m)$, since otherwise M cannot work conservatively. Finally, $L(\hat{G}_z) = L(G_m)$ and, by assumption, $\hat{T}_j \subseteq L(\hat{G}_z)$. Thus $\{s_1, ..., s_n\} \not\subseteq \hat{T}_j$. This proves the claim.

With the following claim we complete the proof of Property (3).

Claim B. If $L(\hat{G}_z) \subset L(\hat{G}_j)$, then there exists an initial segment of some language $L \in \mathcal{L}$ on which M does not work conservatively.

Since $\hat{T}_j \subseteq L(\hat{G}_z) \subset L(\hat{G}_j)$ and, by Claim A, $M(s_1, ..., s_u) \neq m$ for all $u \leq n$ we may conclude that $s_1, ..., s_n$ is an initial segment of the canonical text of $L(\hat{G}_z)$. On the other

hand, the initial segment of the canonical text t^m of $L(\hat{G}_z)$ has to coincide with the canonical text τ^r of some language $L_r \in \mathcal{L}$ up to length y, since $z = \langle m, y \rangle$. Let $s_1, ..., s_n, s_{n+1}, ..., s_y$ be that segment. By construction, there should be a $w \leq y$ such that $M(s_1, ..., s_w) = m$. Due to Claim A, $w > n$. Therefore, when fed successively with $s_1, ..., s_n, s_{n+1}, ..., s_w$ the IIM M outputs k (on input $s_1, ..., s_n$) and in some subsequent step m (on input $s_1, ..., s_w$). Again, taking into account that M works consistently, we get $\{s_1, ..., s_w\} \subseteq L(\hat{G}_z) \cap L_r$. Since $L(G_k) = L(\hat{G}_j) \supset L(\hat{G}_z) = L(G_m)$, we see that the mind change from k to m cannot be caused by a detected inconsistency. Therefore, on the particular initial segment $s_1, ..., s_w$ of some language $L_r \in \mathcal{L}$ the IIM M does not work conservatively.

Sufficiency: It suffices to prove that there is an IIM M inferring any $L \in \mathcal{L}$ on any text with respect to $\hat{\mathcal{G}}$. However, we show show the slightly stronger result $L(\hat{\mathcal{G}}) \subseteq EWMON(M)$ for some IIM M. So let $L(\hat{G}_z) \in L(\hat{\mathcal{G}})$ and let $t \in text(L(\hat{G}_z))$. We define:

$M(t_x) =$ "If $x = 1$ or M on t_{x-1} does not output a hypothesis, then goto (B). Otherwise, goto (A).

(A) Let j be the hypothesis produced last by M when fed with t_{x-1}. Test whether $t_x^+ \subseteq L(\hat{G}_j)$. In case it is, output j and request the next input. Otherwise, goto (B).

(B) For $j = 1, ..., x$, generate $\hat{T}_j$ and test whether $\hat{T}_j \subseteq t_x^+ \subseteq L(\hat{G}_j)$. In case there is one j fulfilling the test, output the minimal one, and request the next input. Otherwise, output nothing and request the next input."

Since all of the $\hat{T}_j$ are uniformly recursively generable and finite, we see that M is an IIM. Moreover, the IIM M changes its mind if and only it finds an inconsistency in (A). Hence, M works weak-monotonically provided it converges on t to a correct hypothesis.

Claim 1. M converges on $(t_x)_{x \in N}$.

Let $k = \mu r[L(\hat{G}_z) = L(\hat{G}_r)]$. Consider $\hat{T}_1$, ..., $\hat{T}_k$. Since $t \in text(L(\hat{G}_z))$, there must be an x such that $\hat{T}_k \subseteq t_x^+ \subseteq L(\hat{G}_k)$. That means, at least after having fed t_x to M, the machine M outputs an hypothesis. Furthermore, since $\hat{T}_k \subseteq t_{x+y}^+ \subseteq L(\hat{G}_k)$ for all $y \in \mathbb{N}$, the IIM M never produces a guess $j > k$. Finally, M changes its mind if and only if it receives some string misclassifying its current guess. Thus, any hypothesis rejected once is never repeated in some subsequent step. Since at least k can never be rejected, M has to converge.

Claim 2. If M converges, say to j, then $L(\hat{G}_z) = L(\hat{G}_j)$.

Suppose the converse, i.e., $M(t_x)$ converges to j and $L(\hat{G}_z) \neq lhj$.

Case 1. $L(\hat{G}_z) \setminus L(\hat{G}_j) \neq \emptyset$.
Consequently, there is at least one string $s \in L(\hat{G}_z) \setminus L(\hat{G}_j)$ that has to appear eventually in t, say in t_x for some x. Thus, $t_x^+ \not\subseteq L(\hat{G}_j)$, a contradiction.

Case 2. $L(\hat{G}_j) \setminus L(\hat{G}_z) \neq \emptyset$.
Then we may restrict ourselves to the case $L(\hat{G}_z) \subset L(\hat{G}_j)$, since otherwise we are again in Case 1. Since $\hat{T}_j \neq \emptyset$ as well as $\hat{T}_j \subseteq t_x^+ \subseteq L(\hat{G}_z)$ for all sufficiently large x, we immediately

obtain $\hat{T}_j \subseteq L(\hat{G}_z)$. Applying Property (3) we may conclude $L(\hat{G}_z) \not\subseteq L(\hat{G}_j)$, a contradiction. Consequently, M converges to j.

q.e.d.

As an immediate consequence of the proof given above we obtain the following corollary.

Corollary 14. *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.

Next, as we promised at the beginning of Section 3, our results may be applied to solve the question whether conservativeness and semantical finiteness are of different impact in language learning. Let us elaborate this topic here. First, Theorem 2 in particular shows that $CWMON \subset CLIM$. Therefore, the indexed family $\mathcal{L}_{ewd}$ cannot be learnt conservatively. On the other hand, we have the following theorem.

Theorem 15. *Let $\mathcal{L}_{ewd}$ be the indexed family defined in the proof of Theorem 2. Then we have:*

There are a hypothesis space $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ and a semantically finite working IIM M such

that M LIM–infers $\mathcal{L}_{ewd}$ w.r.t. $\mathcal{G}$.

Nevertheless, it remained open whether or not any inferable indexed family is semantically finite learnable, too.

Beside this, several other problems are still unsolved. First, it would be very interesting to know whether Corollary 14 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] Brandt, U. (1986), The position of index sets of identifiable sets in the arithmetical hierarchy, *Information and Control* **68**, 185 - 195.

[5] 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.

[6] 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.

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

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

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

[10] 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.

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

[12] 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.

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

[14] 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.

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

[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] SALOMAA, R. (1973), "Formal Languages," Academic Press, London.

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

[24] 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.

[25] 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.