Bulletin of Informatics and Cybernetics, Vol. 27, No. 1, 1995

Trading Monotonicity Demands versus Efficiency

By

Steffen LANGE* **and Thomas ZEUGMANN†**

Abstract

The present paper deals with the learnability of indexed families $\mathcal{L}$ of uniformly recursive languages *from positive data*. We consider the influence of three monotonicity demands and their dual counterparts to the efficiency of the learning process. The efficiency of learning is measured in dependence on the number of mind changes a learning algorithm is allowed to perform. The three notions of (dual) monotonicity reflect different formalizations of the requirement that the learner has to produce better and better (specializations) generalizations when fed more and more data on the target concept.

We distinguish between *exact* learnability ($\mathcal{L}$ has to be inferred with respect to $\mathcal{L}$), *class preserving* learning ($\mathcal{L}$ has to be inferred with respect to some suitably chosen enumeration of all the languages from $\mathcal{L}$), and *class comprising* inference ($\mathcal{L}$ has to be learned with respect to some suitably chosen enumeration of uniformly recursive languages containing at least all the languages from $\mathcal{L}$).

In particular, we prove that a relaxation of the relevant (dual) monotonicity requirement may result in an arbitrarily large speed-up. However, whether or not such a speed-up may be achieved crucially depends on the set of allowed hypothesis spaces as well as of the (dual) monotonicity demands involved.

1. Introduction

Inductive inference is the process of hypothesizing a general rule from eventually incomplete data. Within the last three decades it received much attention from computer scientists. Nowadays inductive inference can be considered as a form of machine learning with potential applications to artificial intelligence (cf., e.g., Angluin and Smith [3, 4], Osherson, Stob and Weinstein [36]). For more information concerning recent developments in inductive inference, the reader is referred to the annual Workshops on Computational Learning Theory, COLT (cf., e.g., Rivest et al. [37], Fulk and Case [15], Haussler [18], and Warmuth [39]), the International Workshops on Algorithmic Learning Theory, ALT (cf., e.g., Arikawa et al. [5, 6, 7]) and the International Workshops

* HTWK Leipzig, Fachbereich Mathematik und Informatik, PF 66, 04251 Leipzig, Germany, Email: steffen@informatik.th-leipzig.de

† Research Institute of Fundamental Information Science, Kyushu University 33, Fukuoka 812, Japan, Email: thomas@rifis.kyushu-u.ac.jp

on Analogical and Inductive Inference, AII (cf., e.g., Jantke [20, 22], and Arikawa and Jantke [7]).

The present paper deals with inductive inference of formal languages, a field in which many interesting and sometimes surprising results have been obtained (cf., e.g., Case and Lynes [12], Case [11], Fulk [14]). Looking at potential applications, Angluin [1, 2] started the systematic study of learning enumerable families of uniformly recursive languages, henceforth called *indexed families*. Recently, this topic has attracted much attention (cf., e.g., [23, 24, 25, 27, 28, 29, 34, 38, 43]).

Next we specify the information from which the target languages have to be learned. A *text* of a language L is an infinite sequence of strings that eventually contains all strings of L. Alternatively, one can consider learning from *informant*. An informant of a language L is an infinite sequence of all strings over the underlying alphabet that are classified with respect to their containment in L. In this paper we exclusively deal with the learnability from text.

An algorithmic learner, henceforth called *inductive inference machine* (abbr. IIM), takes as input initial segments of a text, and outputs, from time to time, a hypothesis about the target language. The set $\mathcal{G}$ of all admissible hypotheses is called *hypothesis space*. Furthermore, the sequence of hypotheses has to converge to a hypothesis correctly describing the language to be learned, i.e., after some point, the IIM stabilizes to an accurate hypothesis. If there is an IIM that learns a language L from all texts for it, then L is said to be *learnable in the limit from text* with respect to the hypothesis space $\mathcal{G}$. Consequently, when dealing with learning in the limit, we are faced with an ongoing inference process. If $d_0, ..., d_x$, $x = 0, 1, 2, ...$, denotes the sequence of data the IIM M is successively fed, then we use j_x to denote the last hypothesis output by M, if any, on successive input $d_0, ..., d_x$. We say that M changes its mind, or synonymously, M performs a mind change, iff $j_x \neq j_{x+1}$. The number of mind changes is a measure of efficiency and has been introduced by Barzdin and Freivalds [8]. Subsequently, this measure of efficiency has been intensively studied (cf., e.g., Barzdin et al. [9], Case and Smith [13], Wiehagen et al. [41], Gasarch and Velauthapillai [16]). However, all the mentioned papers considered the learnability of recursive functions. Hence, it is only natural to ask whether or not this measure of efficiency is of equal importance in the setting of language learning. This is indeed the case as recently obtained results show (cf., e.g., Mukouchi [34, 35], Lange and Zeugmann [29], Lange [26]).

In this paper we study problems of higher granularity. In order to explain them we have to describe the monotonicity constraints and their dual counterparts we are going to deal with. The three notions of monotonicity reflect different formalizations of the requirement that the learner has to produce better and better generalizations when fed more and more data on the target concept (cf. Jantke [21], Wiehagen [40]). Interpreting generalization in its strongest sense yields that the learner is forced to produce an augmenting chain of languages, i.e., $L_i \subseteq L_j$ in case L_j is hypothesized later than L_i. This learning model is referred to as *strong monotonic* inference. Restricting "better generalization" to the language L to be learned results in demanding $L_i \cap L \subseteq$

$L_j \cap L$ provided L_j is later guessed than L_i. Learning algorithms behaving thus are called *monotonic*.

Weakening the strong-monotonicity constraint in the same way as the monotonicity principle of classical logic is generalized to cumulativity (cf., e.g., Brewka [10]) yields *weak-monotonic* learning, i.e., now the learner is required to behave strong-monotonically as long as it does not receive data contradicting its actual guess (cf. Definition 2.3).

All the models of monotonic learning share the common property that they model "learning by generalization." Hence, it is only natural to consider their dual counterparts, that is "learning by specialization." These learning models have been introduced by Kapur [23]. The resulting learning types are referred to as *dual strong-monotonic, dual monotonic* and *dual weak-monotonic* learning (cf. Definition 2.4). As Jantke [21] pointed out, the (dual) monotonicity requirements described above reflect different degrees of *non-monotonic* reasoning that may be incorporated into the learning process. However, it is well imaginable that the use of non-monotonic reasoning does not only affect the learnability at all but also the *efficiency* of learning. Kinber [25] first studied this problem for learning recursively enumerable languages. We continue along this line in the setting of uniformly recursive languages.

Clearly, this question is directly related to the problem of what a natural learning algorithm might look like. In particular, it is well imaginable that one may succeed in designing a learning algorithm that fulfills a desirable (dual) monotonicity demand. However, it seems to be interesting to know what price one might have to pay concerning the resulting efficiency. Therefore, we study the influence of different (dual) monotonicity constraints to the number of mind changes an IIM has to perform when inferring a target indexed family. Then, the right question to ask is whether a weakening of the monotonicity requirement may yield a speed-up. Therefore, we always start with a target indexed family inferable under some (dual) monotonicity constraint with an *a priori* fixed number of mind changes. Then we ask whether or not the least or some possible relaxation of the corresponding (dual) monotonicity requirement might help to uniformly reduce the number of mind changes. As we shall see, there is no unique answer to this problem.

2. Preliminaries

Let $\mathbb{N} = \{0, 1, 2, ...\}$ be the set of all natural numbers. We set $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$. Let $\varphi_0,\ \varphi_1,\ \varphi_2, ...$ denote any fixed ***acceptable programming system*** of all (and only all) partial recursive functions over $\mathbb{N}$, and let $\Phi_0,\ \Phi_1,\ \Phi_2, ...$ be any associated ***complexity measure*** (cf. Machtey and Young [33]). Furthermore, let $k, x \in \mathbb{N}$. If $\varphi_k(x)$ is defined (abbr. $\varphi_k(x)\downarrow$) then we also say that $\varphi_k(x)$ converges; otherwise, $\varphi_k(x)$ diverges (abbr. $\varphi_k(x)\uparrow$). By $\langle\cdot,\cdot\rangle\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$ we denote ***Cantor's pairing function***. In the sequel we assume familiarity with formal language theory (cf., e.g., Hopcroft and Ullman [19]). By Σ we denote any fixed finite alphabet of symbols. Let Σ^* be the free monoid over Σ, and let $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$, where ε denotes the empty string.

Any $L \subseteq \Sigma^*$ is called a language. By $co-L$ we denote the complement of L. Let L be a language and $t = s_0, s_1, s_2, \ldots$ an infinite sequence of strings from Σ^* such that $range(t) = \{s_k \mid k \in \mathbb{N}\} = L$. 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 $x \in \mathbb{N}$. Then t_x denotes the initial segment of t of length $x+1$, and t_x^+ its range, i.e., $t_x^+ = \{s_k \mid k \leq x\}$.

Next, we introduce the notion of the ***canonical text*** that turned out to be very helpful in proving several theorems. Let $s_0, s_1, s_2, \ldots$ be the lexicographically ordered text of Σ^*, i.e., all strings over the corresponding underlying alphabet Σ ordered by increasing length, where strings having the same length are lexicographically ordered. Let $L \subseteq \Sigma^*$ be any non-empty recursive language. Then the canonical text of L is obtained as follows. Test sequentially whether $s_z \in L$ for $z = 0, 1, 2, \ldots$ until the first z is found such that $s_z \in L$. Since $L \neq \emptyset$ there must be at least one z fulfilling the test. Set $t_0 = s_z$. We proceed inductively. For all $x \in \mathbb{N}$ we define:

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

Note that the canonical text of every non-empty recursive language can be effectively computed provided an effective procedure deciding membership in L is given.

In this paper we deal with the learnability of indexed families defined as follows: A sequence $L_0, L_1, L_2, \ldots$ is said to be an ***indexed family*** provided all languages L_j are non-empty and membership in L_j is uniformly decidable for all $j \in \mathbb{N}$. Note that the definition of an indexed family includes both, a description for every language L_j, and a particular enumeration of all the languages. In particular, every indexed family is directly connected with the grammars behind the enumerated languages. Consequently, we can consider the indices as compiled grammars (cf. Hopcroft and Ullman [19]).

As in Gold [17], 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, and then it requests the next input string.

At this point we have to clarify what hypothesis space we should choose. Since we exclusively deal with the learnability of indexed families $\mathcal{L} = (L_j)_{j \in \mathbb{N}}$ we always take as hypothesis space an enumerable family of grammars $\mathcal{G} = G_0, G_1, G_2, \ldots$ over the terminal alphabet Σ satisfying $range(\mathcal{L}) \subseteq \{L(G_j) \mid j \in \mathbb{N}\}$, and require that membership in $L(G_j)$ is uniformly decidable for all $j \in \mathbb{N}$ and all $s \in \Sigma^*$. When an IIM outputs a number j, we interpret it to mean that the machine is hypothesizing the grammar G_j. Furthermore, let $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ be any hypothesis space. For notational convenience we use $\mathcal{L}(\mathcal{G})$ to denote $(L(G_j))_{j \in \mathbb{N}}$. Note that $\mathcal{L}(\mathcal{G})$ constitutes itself an indexed family for all hypothesis spaces $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$.

Let t be a text, and $x \in \mathbb{N}$. Then we use $M(t_x)$ to denote the last hypothesis produced by M when successively fed t_x. The sequence $(M(t_x))_{x \in \mathbb{N}}$ is said to ***converge in the limit*** to the number j if and only if either $(M(t_x))_{x \in \mathbb{N}}$ is infinite and all but

finitely many terms of it are equal to j, or $(M(t_x))_{x\in\mathbb{N}}$ is non-empty and finite, and its last term is j. Now we are ready to define learning in the limit from positive data.

DEFINITION 2.1. (GOLD [17]) *Let $\mathcal{L}$ be an indexed family, let L be a language, and let $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ be a hypothesis space.* ***An IIM M $CLIM$–identifies L from text with respect to $\mathcal{G}$*** *iff for every text t for L, there exists a $j \in \mathbb{N}$ such that the sequence $(M(t_x))_{x\in\mathbb{N}}$ converges in the limit to j and $L = L(G_j)$.*

Furthermore, M $CLIM$–identifies $\mathcal{L}$ with respect to $\mathcal{G}$ if and only if, for each $L \in range(\mathcal{L})$, M $CLIM$–identifies L with respect to $\mathcal{G}$.

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

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

By the definition of convergence, whenever an IIM identifies the language L, then it performs at most finitely many mind changes. However, the precise number of mind changes may well vary from text to text as well as for every language $L \in range(\mathcal{L})$. In particular, the number of allowed mind changes is *not* required to be *universally bounded* for all $L \in range(\mathcal{L})$. Within the next definition we consider the special case that the number of allowed mind changes is universally bounded by an *a priori* fixed number.

DEFINITION 2.2. (BARZDIN AND FREIVALDS [8]) *Let $\mathcal{L}$ be an indexed family, let L be a language, let $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ be a hypothesis space, and let $n \in \mathbb{N} \cup \{*\}$.* ***An IIM $CLIM_n$–identifies L from text with respect to $\mathcal{G}$*** *iff*

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

(2) *for every text t for L the IIM M performs, when fed t, at most n ($n = *$ means at most finitely many) mind changes, i.e., $card(\{x \mid M(t_x) \neq M(t_{x+1})\}) \leq n$.*

Moreover, M $CLIM_n$–identifies $\mathcal{L}$ with respect to $\mathcal{G}$ if and only if, for each $L \in range(\mathcal{L})$, M $CLIM_n$–identifies L with respect to $\mathcal{G}$.

$CLIM_n$, LIM_n and $ELIM_n$ are defined in the same way as above.

Obviously, $\lambda LIM_* = \lambda LIM$ for all $\lambda \in \{E, \varepsilon, C\}$. Moreover, λLIM_0 is also referred to as *finite learning* for $\lambda \in \{E, \varepsilon, C\}$, since the IIM is only allowed to produce a single guess that cannot be changed later (cf. Gold [17]). Note that the learning types λLIM_n do heavily depend on $\lambda \in \{E, \varepsilon, C\}$ for all $n \geq 1$ (cf. Lange and Zeugmann [29], Lange

[26]).

The following proposition summarizes the known results concerning learning in the limit from text with respect to the number of allowed mind changes.

PROPOSITION 2.1.

$$\begin{array}{ccccc}
ELIM & = & LIM & = & CLIM \\
\cup & & \cup & & \cup \\
\bigcup_{n\in\mathbb{N}} ELIM_n & \subset & \bigcup_{n\in\mathbb{N}} LIM_n & \subset & \bigcup_{n\in\mathbb{N}} CLIM_n \\
\cup & & \cup & & \cup \\
\vdots & & \vdots & & \vdots \\
\cup & & \cup & & \cup \\
ELIM_3 & \subset & LIM_3 & \subset & CLIM_3 \\
\cup & & \cup & & \cup \\
ELIM_2 & \subset & LIM_2 & \subset & CLIM_2 \\
\cup & & \cup & & \cup \\
ELIM_1 & \subset & LIM_1 & \subset & CLIM_1 \\
\cup & & \cup & & \cup \\
ELIM_0 & = & LIM_0 & = & CLIM_0
\end{array}$$

Note that the hierarchy in the first column has been independently obtained by Mukouchi [34].

Next, we want to formally define strong-monotonic, monotonic and weak-monotonic inference.

DEFINITION 2.3. (JANTKE [21]; WIEHAGEN [40]) *Let L be a language, and let $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ be a hypothesis space.* ***An IIM M is said to identify the language L from text with respect to $\mathcal{G}$***

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

(B) ***monotonically***

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

iff

M CLIM–identifies L with respect to $\mathcal{G}$ and for every text t of 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}, respectively, where $k \geq 1, k \in \mathbb{N}$, the following conditions are satisfied:

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

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

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

We denote by $CSMON$, $CMON$, $CWMON$ the collection of all those indexed families $\mathcal{L}$ for which there are a hypothesis space $\mathcal{G}$ and an IIM inferring them strong-monotonically, monotonically, and weak-monotonically from text with respect to $\mathcal{G}$.

Note that the learning types $\lambda SMON$, λMON, and $\lambda WMON$ do heavily depend on $\lambda \in \{E, \varepsilon, C\}$ (cf. Lange and Zeugmann [30, 31] and Lange, Zeugmann and Kapur [32]).

PROPOSITION 2.2.

$$\begin{array}{ccccc} ELIM & = & LIM & = & CLIM \\ \cup & & \cup & & \cup \\ EWMON & \subset & WMON & \subset & CWMON \\ \cup & & \cup & & \cup \\ EMON & \subset & MON & \subset & CMON \\ \cup & & \cup & & \cup \\ ESMON & \subset & SMON & \subset & CSMON \\ \cup & & \cup & & \cup \\ ELIM_0 & = & LIM_0 & = & CLIM_0 \end{array}$$

Furthermore, we use $CSMON_n$, $CMON_n$, $CWMON_n$, where $n \in \mathbb{N}$, to denote the collections of all those indexed families $\mathcal{L}$ for which there are a hypothesis space $\mathcal{G}$ and an IIM inferring them strong-monotonically, monotonically, and weak-monotonically from text *with at most n mind changes* with respect to $\mathcal{G}$.

We continue in formally defining the three types of dual monotonic language learning mentioned in Section 1.

DEFINITION 2.4. (KAPUR [23]) *Let L be a language, and let $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ be a hypothesis space.* ***An IIM M is said to identify the language L from text with respect to $\mathcal{G}$***

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

(B) ***dual monotonically***

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

iff

M CLIM–identifies L with respect to $\mathcal{G}$ and for every text t of 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}, respectively, where $k \geq 1, k \in \mathbb{N}$, the following conditions are satisfied:

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

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

(C) *if* $t^+_{x+k} \subseteq L(G_{j_x})$, *then* $co\text{-}L(G_{j_x}) \subseteq co\text{-}L(G_{j_{x+k}})$.

By $CSMON^d$, $CMON^d$, and $CWMON^d$ we denote the collections of all those indexed families $\mathcal{L}$ for which there are a hypothesis space $\mathcal{G}$ and an IIM identifying them dual strong-monotonically, dual monotonically and dual weak-monotonically from text with respect to $\mathcal{G}$, respectively.

Note that the learning types $\lambda SMON^d$, λMON^d, and $\lambda WMON^d$ do heavily depend on $\lambda \in \{E, \varepsilon, C\}$ (cf. Lange and Zeugmann [30, 31] and Lange, Zeugmann and Kapur [32]). The following proposition summarizes the results obtained.

PROPOSITION 2.3.

$$\begin{array}{ccccc}
ELIM & = & LIM & = & CLIM \\
\cup & & \cup & & \| \\
EWMON^d & \subset & WMON^d & \subset & CWMON^d \\
\cup & & \cup & & \cup \\
EMON^d & \subset & MON^d & \subset & CMON^d \\
\cup & & \cup & & \cup \\
ESMON^d & = & SMON^d & \subset & CSMON^d \\
\| & & \| & & \cup \\
ELIM_0 & = & LIM_0 & = & CLIM_0
\end{array}$$

Finally, we use $CSMON_n^d$, $CMON_n^d$, $CWMON_n^d$, where $n \in \mathbb{N}$, to denote the collections of all those indexed families $\mathcal{L}$ for which there are a hypothesis space $\mathcal{G}$ and an IIM inferring them dual strong-monotonically, dual monotonically, and dual weak-monotonically from text *with at most n mind changes* with respect to $\mathcal{G}$.

3. Results

In this section we study the problem whether or not any of the monotonicity constraints and their dual counterparts defined above may be traded versus the efficiency of learning. Since each monotonicity and dual monotonicity demand has its peculiarities, we handle each of them separately in a special subsection. Moreover, in the following we exclusively consider the case where at least one mind change is mandatory, since otherwise finite learning is compared with some type of monotonic learning.

3.1. Strong-Monotonic Inference

We start our investigations with the strongest possible monotonicity constraint, i.e., with $SMON$ and its variations.

THEOREM 3.1. *Let $\mathcal{L}$ be an indexed family. Then, for every $n \in \mathbb{N}^+$ we have:*

(1) $\mathcal{L} \in ESMON_{n+1} \setminus ESMON_n$ *implies* $\mathcal{L} \notin CLIM_n$,

(2) $\mathcal{L} \in SMON_{n+1} \setminus SMON_n$ *implies* $\mathcal{L} \notin CLIM_n$.

PROOF. The proof is based on the following observations. Let $\hat{M}$ be any strong-monotonic IIM, and let $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ be any hypothesis space such that $\hat{M}$ witnesses $\mathcal{L} \in SMON_{n+1}$ with respect to $\mathcal{G}$. Then the IIM $\hat{M}$ can be simulated by an IIM M such that for all texts $t \in \bigcup_{L \in range(\mathcal{L})} text(L)$ and all $x \in \mathbb{N}$

(A) if M on input t_x makes an output j_x then $t_x^+ \subseteq L(G_{j_x})$, i.e., M is consistent, and if $M(t_x) \neq M(t_{x+1})$, then $t_{x+1}^+ \not\subseteq L(G_{j_x})$, i.e., M is conservative,

(B) M witnesses $\mathcal{L} \in SMON_{n+1}$ with respect to $\mathcal{G}$, i.e., M performs at most as many mind changes as $\hat{M}$ does (cf. Lange and Zeugmann [28]).

Let $\mathcal{L}$ be any indexed family with $\mathcal{L} \in SMON_{n+1} \setminus SMON_n$. Furthermore, let $\mathcal{L} \in SMON_{n+1}$ be witnessed by M, where M is chosen in accordance with (A) and (B). Since $\mathcal{L} \notin SMON_n$, there have to be an $L \in range(\mathcal{L})$ and a text t for L such that M changes its mind exactly $n+1$ times when fed t. Let $j_0, \ldots, j_{n+1}$ denote the finite sequence of M's mind changes produced on t. Since M is strong-monotonic, consistent and conservative, we directly obtain that $L(G_{j_0}) \subset \cdots \subset L(G_{j_{n+1}}) = L$.

Now, $\mathcal{L} \notin CLIM_n$ is a direct consequence of Proposition 3.7 by Mukouchi [35]. This proves Assertion (2), and the same arguments prove Assertion (1). □

The latter theorem allows the following interpretation. Relaxing the requirement to learn exactly (class preservingly) strong-monotonically as much as possible does not increase the efficiency. This is even true, if we are allowed to choose an arbitrary class comprising hypothesis space provided that the target indexed family is inferable in the sense of $ESMON_{n+1}$ ($SMON_{n+1}$), but cannot be exactly (class preservingly) and strong-monotonically learned with at most n mind changes for some $n \in \mathbb{N}$. Hence, in the case considered in Theorem 3.1 the possible efficiency of learning is completely determined by the *topology* of the target indexed family.

Next we consider the class comprising case. Interestingly enough, now the situation considerably changes. The following theorem shows that a suitable choice of the hypothesis space may increase the efficiency of learning, even under the strong-monotonicity constraint.

THEOREM 3.2. *For every $n \in \mathbb{N}^+$ there exists an indexed family $\mathcal{L}$ such that $\mathcal{L} \in (CSMON_{n+1} \cap ELIM_n) \setminus CSMON_n$.*

PROOF. First we prove the $n = 1$ case. Moreover, we use this case to fully explain the basic *proof technique* developed. The first idea is to incorporate a non-recursive but recursively enumerable problem in the definition of the target indexed family. Note that this incorporation has to be done in a way such that membership in the enumerated languages remains uniformly decidable. For that purpose, we used the halting problem. Without loss of generality, we may assume that $\Phi_j(j) \geq 1$ for all $j \in \mathbb{N}$.

The desired indexed family is defined as follows. For all $k \in \mathbb{N}$, we set $L_{3k} = \{a^k b^z \mid z \in \mathbb{N}^+\}$. By convention, a^0 equals the empty string. The remaining languages are defined via their characteristic functions. Let $k \in \mathbb{N}$ and let s denote any string in $\{a, b, c, d\}^*$.

$$f_{3k+1}(s) = \begin{cases} 1, & \text{if } s = a^k b^m \text{ for some } m \in \mathbb{N} \text{ and } \neg\, \Phi_k(k) \leq m, \\ 1, & \text{if } s = a^k c^m \text{ for some } m \in \mathbb{N} \text{ and } \Phi_k(k) = m, \\ 0, & \text{otherwise.} \end{cases}$$

$$f_{3k+2}(s) = \begin{cases} 1, & \text{if } s = a^k b^m \text{ for some } m \in \mathbb{N}, \\ 1, & \text{if } s = a^k d^m \text{ for some } m \in \mathbb{N} \text{ and } \Phi_k(k) = m, \\ 0, & \text{otherwise.} \end{cases}$$

Since the predicate "$\Phi_i(x) \leq y$" is uniformly decidable for all $i, x, y \in \mathbb{N}$, it is easy to verify that $\mathcal{L} = (L_j)_{j \in \mathbb{N}}$ is an indexed family. Next, we present an alternative definition which gives some more insight into the topological structure of $\mathcal{L}$.

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

Then we set $L_{3k+1} = L_{3k+2} = L_{3k}$.

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

Then, let $m = \Phi_k(k)$. We set $L_{3k+1} = \{a^k b^z \mid 1 \leq z \leq m\} \cup \{a^k c^m\}$, and

$L_{3k+2} = L_{3k} \cup \{a^k d^m\}$.

Whenever $\varphi_k(k) \downarrow$, the main problem for any strong-monotonic IIM consists in learning the finite language L_{3k+1} with at most one mind change. Hence, for proving $\mathcal{L} \in CSMON_2$, another ingredient is required, i.e., a suitable choice of a hypothesis space (cf. Claim 1). The harder part is to show that $\mathcal{L} \notin CSMON_1$. As long as only class preserving hypothesis spaces are allowed, it is intuitively obvious that any IIM M strong-monotonically learning $\mathcal{L}$ has to solve the halting problem. However, we have additionally to show that no choice of the hypothesis space may prevent M to recursively handle the halting problem. This part of the proof exploits to a larger extend the assumption that membership is uniformly decidable (cf. Claim 3).

We continue with the formal proof.

Claim 1. $\mathcal{L} \in CSMON_2$.

First of all we define a suitable hypothesis space $\hat{\mathcal{L}} = (\hat{L}_j)_{j \in \mathbb{N}}$. For all $k \in \mathbb{N}$, we set $\hat{L}_{4k} = L_{3k}$, $\hat{L}_{4k+1} = L_{3k+1}$, $\hat{L}_{4k+2} = L_{3k+2}$, and $\hat{L}_{4k+3} = L_{3k} \cap L_{3k+1}$.

Taking into consideration that $\mathcal{L}$ constitutes an indexed family it is not hard to see that $\hat{\mathcal{L}}$ is an indexed family as well. Now we define an IIM M which strong-monotonically identifies $\mathcal{L}$ with respect to $\hat{\mathcal{L}}$.

Let $L \in range(\mathcal{L})$, let t be any text for L, and let $x \in \mathbb{N}$.

IIM *M*: "On input t_x do the following: Determine the unique k such that $a^k b^z \in t_x^+$ for some $z \in \mathbb{N}$. Test whether or not $t_x^+ \subseteq \hat{L}_{4k+3}$. In case it is, output $4k+3$, and request the next input.
Otherwise, goto (A).

(A) Compute $m = max\{m \mid a^k b^m \in \hat{L}_{4k+3}\}$. In case that $a^k c^m \in t_x^+$, output $4k+1$, and request the next input.
Otherwise, goto (B).

(B) If $a^k d^m \in t_x^+$, then output $4k+2$, and request the next input.
Else, output $4k$, and request the next input."

Now, one easily verifies that M $CSMON_2$–learns $\mathcal{L}$. This proves Claim 1.

Claim 2. $\mathcal{L} \in ELIM_1$.

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

IIM ***M*:** "On input t_x do the following: Determine the unique k such that $a^k b^z \in t_x^+$ for some $z \in \mathbb{N}$. Test whether or not $t_x^+ \subseteq L_{3k}$. In case it is, output $3k$, and request the next input.

Otherwise, goto (A).

(A) Compute $m = \Phi_k(k)$. In case that $a^k c^m \in t_x^+$, output $3k+1$, and request the next input.

Otherwise, output $3k+2$, and request the next input."

It remains to show that M $ELIM_1$–infers $\mathcal{L}$. By construction, if M performs a mind change, then it has detected an inconsistency. But in accordance with the definitions of $\mathcal{L}$ and M, $t_x^+ \not\subseteq L_{3k}$ can happen if and only if $\Phi_k(k)$ is defined. Hence, the IIM may compute $m = \Phi_k(k)$. By construction, only two cases are possible, i.e., either L contains $a^k c^m$ or it comprises $a^k d^m$. Looking at the definitions of M and $\mathcal{L}$ it directly follows that M's second guess is correct. Hence, M $ELIM_1$–infers $\mathcal{L}$. This proves Claim 2.

Claim 3. $\mathcal{L} \notin CSMON_1$.

Suppose, there are a class comprising hypothesis space $\mathcal{G}$ for $\mathcal{L}$, and an IIM M witnessing $\mathcal{L} \in CSMON_1$ with respect to $\mathcal{G}$. Then M may be used to design an effective procedure solving the halting problem for the acceptable programming system $\varphi_0, \varphi_1, \ldots$. This can be seen as follows.

Procedure H

"Let $k \in \mathbb{N}$, and let t be the canonical text for L_{3k}. For $x = 0, 1, \ldots$, compute $M(t_x)$ until the minimal index z is found such that M, on successive input t_z outputs its first guess, say j. Test whether or not $\Phi_k(k) \leq z+1$. In case it is, output $\varphi_k(k)\downarrow$. Otherwise, output $\varphi_k(k)\uparrow$."

First, we show that H is an effective procedure. In particular, M has to infer L_{3k} from t. Hence, there is a z such that M on input t_z computes a hypothesis j. Thus, H is recursive and terminates for all $k \in \mathbb{N}$.

It remains to show that H correctly works. Obviously, if the output is $\varphi_k(k)\downarrow$, then $\varphi_k(k)$ is indeed defined. Suppose, H outputs $\varphi_k(k)\uparrow$ but $\varphi_k(k)$ is defined. Hence, $\Phi_k(k)$ is defined, too. Let $m = \Phi_k(k)$. By construction, $m > z+1$. Since M is a strong-monotonic IIM, one easily verifies that $L(G_j) \notin range(\mathcal{L})$. Furthermore, M has to infer L_{3k} from its canonical text. Hence, there has to be a $y > z$ such that $M(t_y) = r$ and $L(G_r) = L_{3k}$. Therefore, M performs at least one mind change when seeing t_y. Finally, due to our construction, $\varphi_k(k)\downarrow$ implies $L_{3k+2} = L_{3k} \cup \{a^k d^m\}$. Consequently, t_y may be extended to a text for L_{3k+2} on which M has to perform an additional mind change, a contradiction.

The cases $n > 1$ may be proved using the same "lifting" technique as in Lange and Zeugmann [29] (cf. the proof of Theorem 3.3 below for more details). □

At this point it is only natural to ask whether the latter theorem generalizes to all indexed families from $CSMON_{n+1} \setminus CSMON_n$ not belonging to $SMON$. The negative answer is provided by our next theorem.

THEOREM 3.3. *For all $n \in \mathbb{N}$, there exists an indexed family $\mathcal{L}$ such that*

(1) $\mathcal{L} \in CSMON_{n+1} \setminus SMON$,

(2) $\mathcal{L} \notin CLIM_n$.

PROOF. We consider the following indexed family $\mathcal{L}_{sep_0} = (L_j)_{j \in \mathbb{N}}$. For all $k \in \mathbb{N}$, we define $L_{2k} = \{a^k b^j \mid j \in \mathbb{N}^+\}$ and $L_{2k+1} = \{a^k b^j \mid 1 \leq j \leq \Phi_k(k)\} \cup \{c^{\Phi_k(k)}\}$. Note that the set $\{c^{\Phi_k(k)}\}$ is defined to be empty, if $\varphi_k(k)$ diverges. Then it is easy to show that $\mathcal{L}_{sep_0} \in CSMON_1 \setminus SMON$. $\mathcal{L}_{sep_0} \in CSMON_1$ can be straightforwardly verified, if one selects the following hypothesis space $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$. For all $k \in \mathbb{N}$, set $L(G_{3k}) = L_{2k}$, $L(G_{3k+1}) = L_{2k+1}$, and $L(G_{3k+2}) = L_{2k} \cap L_{2k+1}$. Furthermore, an easy application of the idea underlying the proof of Claim 3 in Theorem 3.2 yields $\mathcal{L}_{sep_0} \notin SMON$. Since $CLIM_0 \subset SMON$ (cf. Proposition 2.2), $\mathcal{L}_{sep_0}$ cannot be finitely inferred. This proves the $n = 0$ case.

Again, an application of the same "lifting" technique as in Lange and Zeugmann [29] yields the theorem for all $n \in \mathbb{N}$. Let us briefly describe the underlying construction. First we separate $CSMON_2$ and $CLIM_1$ via a family $\mathcal{L}_{sep1}$ which does not belong to $SMON$. Thereby, we explain the basic idea. Then we describe how the construction can be generalized.

Let $\mathcal{L}_{sep_0}$ be the indexed family introduced above. For any $L_m \in \mathcal{L}_{sep_0}$, $m \in \mathbb{N}$, we define $\tilde{L}_m = L_m \cup \{d\}$. Moreover, let $\mathcal{L}_{sep_1}$ be any canonical enumeration of all the languages L_m and $\tilde{L}_m$. Similarly to the $n = 0$ case, $\mathcal{L}_{sep_1} \in CSMON_2 \setminus SMON$ can be easily verified. We omit the details. It remains to prove that $\mathcal{L}_{sep_1} \notin CLIM_1$. This is indirectly done via the following claim.

Claim 1. If $\mathcal{L}_{sep_1} \in CLIM_1$, then $\mathcal{L}_{sep_0} \in CLIM_0$.

Let M be any IIM and $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ any hypothesis space such that M $CLIM_1$–identifies $\mathcal{L}_{sep_1}$ with respect to $\mathcal{G}$. Furthermore, let $\tilde{L}_m = L_m \cup \{d\}$ be arbitrarily fixed, and let t be any text for L_m. Since M has to infer L_m too, there must be a z such that $M(t_x) = j$ for all $x \geq z$ and $L_m = L(G_j)$. But now we may feed the following text $\tilde{t}$ for $\tilde{L}_m$ to M. First, we input the initial segment t_z, then the string d, and subsequently all the remaining strings of t. By construction, $M(t_z) = j$ but $\tilde{L}_m \neq L(G_j)$. Consequently, M has to perform one more mind change. That means, on t the machine has never changed its mind, since otherwise M does not $CLIM_1$–identifies $\mathcal{L}_{sep_1}$. Hence, M witnesses $\mathcal{L}_{sep_0} \in CLIM_0$. This proves Claim 1.

On the other hand, as we have already seen, there is no IIM finitely inferring $\mathcal{L}$. Thus, by contraposition of Claim 1 we directly obtain that $\mathcal{L}_{sep_1} \notin CLIM_1$. This proves the theorem in case $n = 1$.

The general case is handled by induction. Suppose, we have already separated $CSMON_{n+1}$ and $CLIM_n$ in using the indexed family $\mathcal{L}_{sep_n}$ with $\mathcal{L}_{sep_n} \notin SMON$. For any $m \in \mathbb{N}$, we set $\tilde{L}_m = L_m \cup \{d, d^2, \ldots, d^{n+1}\}$ and define $\mathcal{L}_{sep_{n+1}}$ to be any canonical enumeration of all the languages from $\mathcal{L}_{sep_n}$ and of all $\tilde{L}_m$. Again, one straightforwardly

shows that $\mathcal{L}_{sep_{n+1}} \in CSMON_{n+2} \setminus SMON$. Moreover, an easy modification of the demonstration above directly yields the following claim.

Claim 2. If $\mathcal{L}_{sep_{n+1}} \in CLIM_{n+1}$, then $\mathcal{L}_{sep_n} \in CLIM_n$.

Finally, the latter claim immediately contradicts the induction hypothesis that $\mathcal{L}_{sep_n} \notin CLIM_n$. Thus, by contraposition of Claim 2 one obtains $\mathcal{L}_{sep_{n+1}} \notin CLIM_{n+1}$. We omit the details. □

Note that the proof of the latter theorem directly allows the following corollary which sharpens the assertion $SMON \subset CSMON$ pointed out in Proposition 2.2.

COROLLARY 3.4. $CSMON_1 \setminus SMON \neq \emptyset$.

Theorem 3.3 directly yields the problem whether or not Theorem 3.2 can be strengthened, i.e., whether or not the number of mind changes that can be traded versus the strong-monotonicity constraint is bounded by one. The answer is provided by our next theorem.

THEOREM 3.5. *For every $n \in \mathbb{N}^+$ there exists an indexed family $\mathcal{L}$ such that*

$\mathcal{L} \in (CSMON_{n+1} \cap EMON_1) \setminus CSMON_n$.

PROOF. Again, we only sketch the proof using the $n = 2$ case, thereby explaining the *proof technique* developed. The main idea is to suitably iterate the proof technique presented in the demonstration of Theorem 3.2. Therefore, we incorporate one more halting problem into the definition of the indexed family $\mathcal{L}$ witnessing $\mathcal{L} \in CSMON_3 \setminus CSMON_2$, and $\mathcal{L} \in EMON_1$. This is done as follows. Without loss of generality, we may assume that $\Phi_j(j) \geq 1$ for all $j \in \mathbb{N}$. We define:

$L_{4\langle k_1,k_2\rangle} = \{a^{\langle k_1,k_2\rangle}b^z \mid z \in \mathbb{N}^+\}$ for all $k_1, k_2 \in \mathbb{N}$. The remaining languages of $\mathcal{L}$ are defined as follows.

Case 1. $\varphi_{k_1}(k_1)\uparrow$

Then $L_{4\langle k_1,k_2\rangle+1} = L_{4\langle k_1,k_2\rangle+2} = L_{4\langle k_1,k_2\rangle+3} = L_{4\langle k_1,k_2\rangle}$.

Case 2. $\varphi_{k_1}(k_1)\downarrow$ and $\varphi_{k_2}(k_2)\uparrow$

Then, let $\ell = \Phi_{k_1}(k_1)$. We set $L_{4\langle k_1,k_2\rangle+1} = \{a^{\langle k_1,k_2\rangle}b^z \mid 1 \leq z \leq \ell\} \cup \{a^{\langle k_1,k_2\rangle}c^\ell\}$, and $L_{4\langle k_1,k_2\rangle+2} = L_{4\langle k_1,k_2\rangle+3} = L_{4\langle k_1,k_2\rangle}$.

Case 3. $\varphi_{k_1}(k_1)\downarrow$ and $\varphi_{k_2}(k_2)\downarrow$

Then, let $\ell = \Phi_{k_1}(k_1)$, and $m = \Phi_{k_2}(k_2)$.

We set $L_{4\langle k_1,k_2\rangle+1} = \{a^{\langle k_1,k_2\rangle}b^z \mid 1 \leq z \leq \ell\} \cup \{a^{\langle k_1,k_2\rangle}c^\ell\}$,

$L_{4\langle k_1,k_2\rangle+2} = \{a^{\langle k_1,k_2\rangle}b^z \mid 1 \leq z \leq \ell + m\} \cup \{a^{\langle k_1,k_2\rangle}d^{\ell+m}\}$, and

$L_{4\langle k_1,k_2\rangle+3} = L_{4\langle k_1,k_2\rangle} \cup \{a^{\langle k_1,k_2\rangle}e^{\ell+m}\}$.

Now, it is easy to see that $\mathcal{L} = (L_z)_{z\in\mathbb{N}}$ constitutes an indexed family. It remains to show that $\mathcal{L}$ fulfills the stated requirements. As in the proof of Theorem 3.2 one proves *mutatis mutandis* that $\mathcal{L} \in EMON_1$, and $\mathcal{L} \in CSMON_3$.

The remaining part, i.e., $\mathcal{L} \notin CSMON_2$, is much harder to prove. For that purpose

we need some additional insight into the behavior of every IIM M that learns $\mathcal{L}$. In particular, we are mainly interested in knowing how M works when successively fed the text $t = a^{\langle k_1,k_2\rangle}b, a^{\langle k_1,k_2\rangle}b^2, a^{\langle k_1,k_2\rangle}b^3, \cdots$ for $L_{4\langle k_1,k_2\rangle}$. That is, t contains all strings from $L_{4\langle k_1,k_2\rangle}$ ordered by increasing length. Since $L_{4\langle k_1,k_2\rangle}$ is infinite, t indeed constitutes a text for it. Moreover, this text is effectively computable. Furthermore, we refer to t as to the *lexicographically ordered text*, since the order in which the strings are enumerated coincides with the lexicographical order of all strings from $L_{4\langle k_1,k_2\rangle}$. The desired information is provided by the following lemma.

LEMMA 3.1. *Let $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ be any class comprising hypothesis space for $\mathcal{L}$ and let M be any IIM witnessing $\mathcal{L} \in CLIM$ with respect to $\mathcal{G}$. 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) $\Phi_{k_1}(k_1) > x + 1$ *and* $\varphi_{k_1}(k_1)\downarrow$,

where t is the lexicographically ordered text of $L_{4\langle k_1,k_2\rangle}$.

Suppose the converse. Then there is a k_2 such that for all k_1, x, j we have: $M(t_x) = j$ implies $\Phi_{k_1}(k_1) \leq x + 1$ or $\varphi_{k_1}(k_1)\uparrow$. 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) is fulfilled, then either $\Phi_{k_1}(k_1) \leq x + 1$ or $\varphi_{k_1}(k_1)\uparrow$. 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 $L_{4\langle k_1,k_2\rangle}$ and simulate M until the first hypothesis j is produced.
Let x_0 be the least x such that $M(t_x) = j$.

(A2) Test whether $\Phi_{k_1}(k_1) \leq x_0 + 1$.
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 infer $L_{4\langle k_1,k_2\rangle}$ from its lexicographically ordered text t. Hence, M should eventually output a hypothesis j when fed t. Furthermore, Instruction (A2) can be effectively accomplished, 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. Since the halting problem is algorithmically undecidable, Lemma 3.1 follows.

LEMMA 3.2. $\mathcal{L} \notin CSMON_2$.

Suppose the converse, i.e., there exist a hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ and an IIM M that $CSMON_2$–learns $\mathcal{L}$ with respect to $\mathcal{G}$. Then we can prove the following lemma.

LEMMA 3.3. *Given any hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ and any program for M witnessing $\mathcal{L} \in CSMON_2$, one can effectively construct an algorithm deciding the halting problem.*

Let $K = \{k \mid \varphi_k(k)\downarrow\}$ and let $j_0, j_1, j_2, ...$ be any fixed effective enumeration of K. We define an algorithm $\mathcal{B}$ as follows.

Algorithm $\mathcal{B}$: "On input k_2 execute (B1) and (B2).

(B1) For $z = 0, 1, 2, ...$ successively compute the lexicographically ordered texts t^{j_0}, t^{j_1}, $t^{j_2}, ..., t^{j_z}$ for $L_{4\langle j_0,k_2\rangle}$, $L_{4\langle j_1,k_2\rangle}, ..., L_{4\langle j_z,k_2\rangle}$ of length $z+1$, respectively. Then, dovetail the simulation of M on successive input of each of these initial segments until the first initial segment $t^{j_r}_x$ $(r, x \leq z)$ and the first hypothesis j are found such that

(α1) $M(t^{j_r}_x) = j$,

(α2) $\Phi_{j_r}(j_r) > x + 1$.

(* By Lemma 3.1, the execution of (B1) has to terminate. *)

(B2) Let $f = \langle j_r, k_2\rangle$ and $\ell = \Phi_{j_r}(j_r)$. Furthermore, we define $\hat{t}_{\ell+y}$ as follows:
$\hat{t}_{\ell+y} = \underbrace{a^f b,\ ...,\ a^f b^{x+1},\ ...,\ a^f b^{\ell}}_{=t^{j_r}_{\ell-1}},\ a^f b^{\ell},\ \underbrace{a^f b^{\ell+1},\ ...,\ a^f b^{\ell+y}}_{y-\text{strings}}$.

For $y = 0, 1, 2, ...$ execute in parallel (β1) and (β2) until (β3) or (β4) happens.

(β1) Test whether $\Phi_{k_2}(k_2) \leq \ell + y$.

(β2) Compute $j_{\ell+y} = M(\hat{t}_{\ell+y})$.

(β3) $\Phi_{k_2}(k_2) \leq \ell + y$ is verified. Then output "$\varphi_{k_2}(k_2)\downarrow$."

(β4) In (β2) a hypothesis $j_{\ell+y} = M(\hat{t}_{\ell+y})$ is computed such that $a^f b^{\ell+1} \in L(G_{j_{\ell+y}})$. Then output "$\varphi_{k_2}(k_2)\uparrow$" and stop."

It remains to show that $\mathcal{B}$ terminates on every input, and behaves correctly.

Claim 1. On every input k_2, the algorithm $\mathcal{B}$ terminates.

As we have already mentioned, by Lemma 3.1 we know that the execution of (B1) has to terminate. Hence, it suffices to show that either (β3) or (β4) happens. Suppose, (β3) does not happen. Then, for all $y \in \mathbb{N}$ we have $\neg\Phi_{k_2}(k_2) \leq \ell + y$. Consequently, $\varphi_{k_2}(k_2)\uparrow$. Therefore, when y tends to infinite, then $\hat{t}_{\ell+y}$ converges to a text for $L_{4\langle j_r,k_2\rangle}$, and hence, M eventually has to output a hypothesis $j_{\ell+y}$ such that $a^f b^{\ell+1} \in L(G_{j_{\ell+y}})$. Thus, ($\beta$4) must happen. This proves Claim 1.

Claim 2. Algorithm $\mathcal{B}$ works correctly.

Obviously, if (β3) happens then $\varphi_{k_2}(k_2)$ is indeed defined. Suppose, (β4) happens. We have to show that $\varphi_{k_2}(k_2)\uparrow$. Suppose the converse, i.e., $\varphi_{k_2}(k_2)\downarrow$. Thus, $\Phi_{k_2}(k_2)$ converges, too. We distinguish the following cases.

Case 1. The hypothesis $j_{\ell+y}$ satisfies $L(G_{j_{\ell+y}}) = L_{4\langle j_r,k_2\rangle}$.

Then M fails to infer $L_{4\langle j_r,k_2\rangle+2}$ strong-monotonically. This can be seen as follows. Since ($\beta 3$) did not happen, we have $\Phi_{k_2}(k_2) > \ell + y$. Hence, $\hat{t}_{\ell+y}$ is an initial segment of a text for $L_{4\langle j_r,k_2\rangle}$ and of a text $\tilde{t}$ for $L_{4\langle j_r,k_2\rangle+2}$. On the other hand, when successively fed $\tilde{t}$, the IIM M sometimes outputs $j_{\ell+y}$, and $L(G_{j_{\ell+y}}) = L_{4\langle j_r,k_2\rangle}$. Since $L_{4\langle j_r,k_2\rangle} \not\subseteq L_{4\langle j_r,k_2\rangle+2}$, we directly see that M violates the strong-monotonicity constraint.

Case 2. The hypothesis $j_{\ell+y}$ does not satisfy $L(G_{j_{\ell+y}}) = L_{4\langle j_r,k_2\rangle}$.

Then, M fails to learn $\mathcal{L}$ with at most two mind changes. Recall that M has already generated the guesses j and $j_{\ell+y}$ when successively fed $\hat{t}_{\ell+y}$. First, we show that $j \neq j_{\ell+y}$. Suppose to the contrary that $j = j_{\ell+y}$. Remember that $\ell = \Phi_{j_r}(j_r)$. Then M fails to infer $L_{4\langle j_r,k_2\rangle+1}$ strong-monotonically. This can be seen as follows. By construction, $a^f b^{\ell+1} \in L(G_{j_{\ell+y}})$, and hence, $a^f b^{\ell+1} \in L(G_j)$. But $j = M(t_x^{j_r})$, and $x < \ell$. Therefore, $t_x^{j_r}$ is an initial segment of some text for $L_{4\langle j_r,k_2\rangle+1}$, too. On the other hand, $a^f b^{\ell+1} \notin L_{4\langle j_r,k_2\rangle+1}$. Consequently, $L(G_j) \not\subseteq L_{4\langle j_r,k_2\rangle+1}$, a contradiction.

Finally, since $j \neq j_{\ell+y}$, M has already performed at least one mind change when successively fed $\hat{t}_{\ell+y}$. Moreover, $\Phi_{k_2}(k_2) = m > \ell + y$, since ($\beta 3$) did not happen. Hence, $\hat{t}_{\ell+y}$ is an initial segment of a text for $L_{4\langle j_r,k_2\rangle}$ as well as for $L_{4\langle j_r,k_2\rangle+3}$. In accordance with $\mathcal{L}$'s definition we additionally have $L_{4\langle j_r,k_2\rangle} \subset L_{4\langle j_r,k_2\rangle+3}$. Thus, we may extend $\hat{t}_{\ell+m}$ with $a^f b^{\ell+y+1}$, $a^f b^{\ell+y+2}$, ... until M learns $L_{4\langle j_r,k_2\rangle}$. This forces M to change its mind again. Afterwards, we present $a^f e^{\ell+m}$, and hence, one more mind change has to occur. Thus, $\mathcal{L} \notin CSMON_2$. This contradiction proves Claim 2. Thus, Lemma 3.3 is shown, and Lemma 3.2 follows. □

Note that the proof of the latter theorem directly allows the following corollary.

COROLLARY 3.6. $EMON_1 \setminus SMON \neq \emptyset$.

3.2. Monotonic Inference

This subsection deals with monotonic inference, and possible relaxations of the monotonicity requirement. But there is a peculiarity which we point out with the following theorem.

THEOREM 3.7. $\lambda LIM_1 = \lambda MON_1$ *for all* $\lambda \in \{E, \varepsilon, C\}$.

PROOF. Here, we discuss exact learning, only. The same idea applies *mutatis mutandis* to handle the remaining cases. By definition $EMON_1 \subseteq ELIM_1$. Now, let $\mathcal{L}$ be an indexed family and M be an IIM which $ELIM_1$–identifies $\mathcal{L}$. It suffices to show that M fulfills the monotonicity constraint as well. In the worst case, M performs at most one mind change, say from j to k, when fed any text t for any $L \in \mathit{range}(\mathcal{L})$. Since M learns L from t, we have $L_k = L$. Therefore, $L_j \cap L \subseteq L_k \cap L = L$ no matter whatever L_j is. □

Next we show that the monotonicity constraint can be traded versus efficiency. This is even true, if the relaxation is as weak as possible, i.e., if the requirement to learn monotonically is relaxed to weak-monotonic inference.

THEOREM 3.8. *For every $n \in \mathbb{N}$, $n \geq 2$, there is an indexed family $\mathcal{L}$ such that $\mathcal{L} \in (EMON_{n+1} \cap EWMON_n) \setminus CMON_n$.*

PROOF. For the sake of presentation, we consider the case $n = 2$. The extension to all $n \geq 3$ may be easily obtained by applying the lifting technique of Lange and Zeugmann [29]. The desired indexed family is defined as follows. Initially, we set $L_0 = \{a\}^+$. For all $k \in \mathbb{N}$, we set $L_{3k+1} = L_0 \cup \{a^k b\}$. In order to define the remaining languages we distinguish the following cases:

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

We set $L_{3k+3} = L_{3k+2} = L_{3k+1}$.

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

Then, let $m = \Phi_k(k)$, and let $\hat{L}_k = \{a^z \mid 1 \leq z \leq m\} \cup \{a^k b\}$.

We set $L_{3k+2} = \hat{L}_k \cup \{a^k c^m\}$, and $L_{3k+3} = \hat{L}_k \cup \{a^k c^m,\ a^k d^m\}$.

After a bit of reflection it is not hard to see that $\mathcal{L} = (L_j)_{j \in \mathbb{N}}$ is an indexed family.

Claim 1. $\mathcal{L} \in EMON_3$.

We define the desired IIM M as follows. Let $L \in range(\mathcal{L})$, let $t \in text(L)$, and let $x \in \mathbb{N}$.

IIM *M*: " On input t_x do the following: As long as $t_x^+ \subseteq L_0$, output the hypothesis 0, and request the next input.
Otherwise, goto (A).

(A) If $a^k b \in t_x^+$ for some $k \in \mathbb{N}$, then determine k and goto (B). Otherwise, repeat the guess 0, and request the next input.

(B) If $a^k d^m \in t_x^+$ for some $m \in \mathbb{N}$, then output $3k + 3$, and request the next input.
If $a^k c^m \in t_x^+$ for some $m \in \mathbb{N}$, then output $3k + 2$.
Otherwise, output $3k + 1$, and request the next input."

Obviously, M monotonically infers $\mathcal{L}$. In the worst case, M changes its mind three times, namely it outputs successively the hypotheses 0, $3k+1$, $3k+2$, and $3k+3$. Since $L_0 \cap L_{3k+2} = L_0 \cap L_{3k+3} \subseteq L_{3k+1} \cap L_{3k+2} = L_{3k+1} \cap L_{3k+3} \subseteq L_{3k+2} \cap L_{3k+3}$, it is easy to verify that each of these mind changes satisfies the monotonicity requirement. This proves Claim 1.

Claim 2. $\mathcal{L} \in EWMON_2$.

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

IIM *M*: "On input t_x do the following: As long as $t_x^+ \subseteq L_0$ output the hypothesis 0, and request the next input.
Otherwise, goto (A).

(A) If $a^kb \in t_x^+$ for some $k \in \mathbb{N}$, then determine k and goto (B). Otherwise, repeat the guess 0, and request the next input.

(B) As long as $t_x^+ \subseteq L_{3k+2}$ output $3k+2$, and request the next input. Otherwise, execute Instruction (C).

(C) If $a^kd^m \in t_x^+$ for some $m \in \mathbb{N}$ output $3k+3$, and request the next input. Otherwise, output $3k+1$, and request the next input."

Obviously, M weak-monotonically infers $\mathcal{L}$. Thereby, M changes its mind at most twice. This proves Claim 2.

Note that the IIM M defined in the latter proof may subsequently output the hypotheses 0, $3k+2$, and $3k+1$ when fed a text for L_{3k+1}. It is easy to verify that the first mind change violates the monotonicity requirement. Moreover, it is easy to argue that $\mathcal{L} \notin EMON_2$. But again, we show a slightly stronger result.

Claim 3. $\mathcal{L} \notin CMON_2$.

Suppose there are a hypothesis space $\mathcal{G}$ and an IIM M such that M $CMON_2$–learns $\mathcal{L}$ with respect to $\mathcal{G}$. By assumption, M, in particular, infers the language L_0 from its text $t = a,\ a^2,\ a^3,\ \ldots$ Thus, there has to be a least index z such that $M(t_z) = j$ and $L(G_j) = L_0$. Given this index z the following recursive predicate ψ solves the halting problem.

Let $k \in \mathbb{N}$; the desired predicate ψ is defined as follows.

$\psi(k) =$ "Test whether or not $\Phi_k(k) \leq z+1$. In case it is, output 1.
Otherwise, execute Instructions (A) and (B).

(A) For $x = 0, 1,\ 2,\ \ldots$ simulate M, when fed
$\hat{t}_{z+1+x} = \underbrace{a,\ \ldots,\ a^{z+1}}_{=t_z},\ a^kb,\ a,\ \ldots,\ a^x$, until the first y is found such that
$j_{z+1+y} = M(\hat{t}_{z+1+y})$ and $a^kb \in L(G_{j_{z+1+y}})$.

(B) Test whether or not $\Phi_k(k) \leq z+1+y$. In case it is, output 1.
Otherwise, output 0."

Obviously, if x tends to infinite then the limit $\hat{t}$ of $\hat{t}_{z+1+x}$ constitutes a text for L_{3k+1}. Since M has to infer the language L_{3k+1} from $\hat{t}$, it is easy to verify that the procedure defined above terminates for every $k \in \mathbb{N}$. Hence, ψ is recursive. It remains to show that $\varphi_k(k)$ is undefined, if $\psi(k) = 0$. Suppose the converse, i.e., $\psi(k) = 0$ as well as $\varphi_k(k)$ is defined. Therefore, $\Phi_k(k) = m > z+1+y$.

Recall that M has already performed at least one mind change when fed $\hat{t}_{z+1+y}$, namely from j to j_{z+1+y}. Since M monotonically infers L_{3k+1} from $\hat{t}$ and $a^kb \in L(G_{j_{z+1+y}})$, we obtain $L(G_{j_{z+1+y}}) \supseteq L_{3k+1}$. Otherwise, M violates the monotonicity constraint when inferring L_{3k+1} from its text $\hat{t}$. Consequently, $L(G_{j_{z+1+y}}) \neq L_{3k+2}$. Now, taking $\mathcal{L}$'s definition into account, it follows that $\hat{t}_{z+1+y}$ may also serve as an initial segment of a text for the language L_{3k+2} because $\Phi_k(k) = m > z+1+y$. Finally,

since $L_{3k+2} \subset L_{3k+3}$, it is easy to verify that $\hat{t}_{z+1+y}$ can be extended to a text for L_{3k+3} such that M has to perform at least two additional mind changes in order to infer L_{3k+3} from this particular text. This contradicts our assumption that M monotonically infers $\mathcal{L}$ with at most two mind changes. Therefore, $\varphi_k(k)$ is undefined, if $\psi(k) = 0$. Consequently, the recursive predicate ψ solves the halting problem for the φ-system. □

Refining *mutatis mutandis* the latter proof analogously as the demonstration of Theorem 3.2 has been extended to show Theorem 3.5, one obtains the following result.

THEOREM 3.9. *For every $n \in \mathbb{N}$, $n \geq 2$, there is an indexed family $\mathcal{L}$ such that*

$\mathcal{L} \in (EMON_{n+1} \cap EWMON_2) \setminus CMON_n$.

PROOF. We present an indexed family which witnesses the desired separation in the $n = 3$ case. This may help to illustrate how the family used in the proof of Theorem 3.8 has to be modified. We define:

$L_0 = \{a\}^+$ and $L_{4\langle k_1,k_2\rangle+1} = L_0 \cup \{a^{\langle k_1,k_2\rangle}b\}$ for all $k_1, k_2 \in \mathbb{N}$. In order to define the remaining languages of $\mathcal{L}$ we distinguish the following cases.

Case 1. $\varphi_{k_1}(k_1) \uparrow$

Then we set $L_{4\langle k_1,k_2\rangle+2} = L_{4\langle k_1,k_2\rangle+3} = L_{4\langle k_1,k_2,j\rangle+4} = L_{4\langle k_1,k_2\rangle+1}$.

Case 2. $\varphi_{k_1}(k_1) \downarrow$ and $\varphi_{k_2}(k_2) \uparrow$

Then, let $\ell = \Phi_{k_1}(k_1)$, and $\hat{L}_{\langle k_1,k_2\rangle} = \{a^z \mid 1 \leq z \leq \ell\} \cup \{a^{\langle k_1,k_2\rangle}b\}$.

We set $L_{4\langle k_1,k_2\rangle+2} = L_{4\langle k_1,k_2\rangle+3} = L_{4\langle k_1,k_2\rangle+4} = \hat{L}_{\langle k_1,k_2\rangle} \cup \{a^{\langle k_1,k_2\rangle}c^\ell\}$.

Case 3. $\varphi_{k_1}(k_1) \downarrow$ and $\varphi_{k_2}(k_2) \downarrow$

Then, let $\ell = \Phi_{k_1}(k_1)$, $m = \Phi_{k_2}(k_2)$, and $\hat{L}_{\langle k_1,k_2\rangle} = \{a^z \mid 1 \leq z \leq \ell\} \cup \{a^{\langle k_1,k_2\rangle}b\}$.

We set $L_{4\langle k_1,k_2\rangle+2} = \hat{L}_{\langle k_1,k_2\rangle} \cup \{a^{\langle k_1,k_2\rangle}c^\ell, a^{\ell+m}\}$,

$L_{4\langle k_1,k_2\rangle+3} = \hat{L}_{\langle k_1,k_2\rangle} \cup \{a^{\langle k_1,k_2\rangle}c^\ell, a^{\langle k_1,k_2\rangle}d^{\ell+m}\}$, and

$L_{4\langle k_1,k_2\rangle+4} = \hat{L}_{\langle k_1,k_2\rangle} \cup \{a^{\langle k_1,k_2\rangle}c^\ell, a^{\langle k_1,k_2\rangle}d^{\ell+m}, a^{\langle k_1,k_2\rangle}e^{\ell+m}\}$.

Again, it is easy to see that $\mathcal{L} = (L_j)_{j\in\mathbb{N}}$ constitutes an indexed family. □

The latter theorems allow the following interpretation. Removing the constraint to learn monotonically may considerably increase the efficiency of the learning process.

Next, we ask whether the overall insight achieved for strong-monotonic and monotonic learning extends to dual strong-monotonic and dual monotonic learning. This is indeed the case as the following two subsections show.

3.3. Dual Strong-Monotonic Learning

First of all recall that $SMON^d = LIM_0 = ELIM_0$ (cf. Proposition 2.3). Consequently, a relaxation of the dual strong-monotonicity constraint may only improve the efficiency of learning, if class comprising hypothesis spaces are admissible. Our next theorem impressively shows that even the weakest possible relaxation, namely allowing to use dual monotonic IIMs instead of dual strong-monotonic machines, may yield a

considerable speed-up.

THEOREM 3.10. *For every $n \in \mathbb{N}^+$ there exists an indexed family $\mathcal{L}$ such that $\mathcal{L} \in (CSMON^d_{n+1} \cap EMON^d_1) \setminus CSMON^d_n$.*

PROOF. It turns out that our main proof technique, namely incorporating a non-recursive but recursively enumerable problem in the definition of the target indexed families, applies *mutatis mutandis* to show the theorem on hand. Therefore, we restrict ourselves to give some insights how to handle the $n = 1$ case.

Without loss of generality, we may assume that $\Phi_j(j) \geq 1$ for all $j \in \mathbb{N}$. For all $k_1, k_2 \in \mathbb{N}$, we set $L_{3\langle k_1,k_2\rangle} = \{a^{\langle k_1,k_2\rangle}b^z \mid z \in \mathbb{N}^+\}$. The remaining languages are defined as follows.

Case 1. $\varphi_{k_1}(k_1)\uparrow$

Then $L_{3\langle k_1,k_2\rangle+1} = L_{3\langle k_1,k_2\rangle+2} = L_{3\langle k_1,k_2\rangle}$.

Case 2. $\varphi_{k_1}(k_1)\downarrow$ and $\varphi_{k_2}(k_2)\uparrow$

Then, let $\ell = \Phi_{k_1}(k_1)$. We set

$L_{3\langle k_1,k_2\rangle+1} = L_{3\langle k_1,k_2\rangle+2} = \{a^{\langle k_1,k_2\rangle}b^z \mid 1 \leq z \leq \ell\} \cup \{a^{\langle k_1,k_2\rangle}c^\ell\}$.

Case 3. $\varphi_{k_1}(k_1)\downarrow$ and $\varphi_{k_2}(k_2)\downarrow$

Then, let $\ell = \Phi_{k_1}(k_1)$, and $m = \Phi_{k_2}(k_2)$.

We set $L_{3\langle k_1,k_2\rangle+1} = \{a^{\langle k_1,k_2\rangle}b^z \mid 1 \leq z \leq \ell\} \cup \{a^{\langle k_1,k_2\rangle}c^\ell, a^{\langle k_1,k_2\rangle}c^{\ell+m}\}$, and

$L_{3\langle k_1,k_2\rangle+2} = \{a^{\langle k_1,k_2\rangle}b^z \mid 1 \leq z \leq \ell\} \cup \{a^{\langle k_1,k_2\rangle}c^\ell, a^{\langle k_1,k_2\rangle}d^{\ell+m}\}$.

Similar to the demonstration of Theorem 3.5, $\mathcal{L} \in EMON^d_1 \setminus CSMON^d_1$ can be easily shown. In order to verify $\mathcal{L} \in CSMON^d_2$ the choice of the underlying hypothesis space is of particular importance. Assume the following hypothesis space $\hat{\mathcal{L}} = (\hat{L}_j)_{j\in\mathbb{N}}$. For all $k_1, k_2 \in \mathbb{N}$, let $\hat{L}_{5\langle k_1,k_2\rangle+z} = L_{3\langle k_1,k_2\rangle+z}$ for all $z \in \{0,1,2\}$, and $\hat{L}_{5\langle k_1,k_2\rangle+3} = \bigcup_{z\in\{0,1,2\}} L_{3\langle k_1,k_2\rangle+z}$, and $\hat{L}_{5\langle k_1,k_2\rangle+4} = \bigcup_{z\in\{1,2\}} L_{3\langle k_1,k_2\rangle+z}$.

Finally, it is not hard to design an IIM M which $CSMON^d_2$–learns $\mathcal{L}$ with respect to $\hat{\mathcal{L}}$. We omit the details. □

3.4. Dual Monotonic Learning

When dealing with dual-monotonic inference we have to take the following peculiarity into consideration. Applying the same idea as in the proof of Theorem 3.7 one directly obtains:

THEOREM 3.11. *$\lambda LIM_1 = \lambda MON^d_1$ for all $\lambda \in \{E, \varepsilon, C\}$.*

Furthermore, removing the requirement to learn dual monotonically may allow the design of IIMs which learn much more efficiently.

THEOREM 3.12. *For every $n \in \mathbb{N}$, $n \geq 2$, there is an indexed family $\mathcal{L}$ such that $\mathcal{L} \in (EMON^d_{n+1} \cap EWMON^d_2) \setminus CMON^d_n$.*

PROOF. Notice that the same families witnessing the corresponding speed-up for

monotonic learning can be used to prove the desired separations. To see this, we investigate the $n = 2$ case in some more detail. Recall the definition of the family $\mathcal{L}$ used in the proof of Theorem 3.8. By definition, $L_0 = \{a\}^+$ and $L_{3k+1} = L_0 \cup \{a^k b\}$ for all $k \in \mathbb{N}$. Moreover, the remaining languages are defined as follows.

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

We set $L_{3k+3} = L_{3k+2} = L_{3k+1}$.

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

Then, let $m = \Phi_k(k)$, and let $\hat{L}_k = \{a^z \mid 1 \leq z \leq m\} \cup \{a^k b\}$.

We set $L_{3k+2} = \hat{L}_k \cup \{a^k c^m\}$, and $L_{3k+3} = \hat{L}_k \cup \{a^k c^m,\ a^k d^m\}$.

Within the proof of Theorem 3.8 we have already shown that $\mathcal{L} \in EWMON_2$. Since every weak-monotonic IIM $\hat{M}$ can be transformed into a dual weak-monotonic IIM which performs at most as many mind changes as $\hat{M}$ does (cf. Lange, Zeugmann and Kapur [32]), it follows $\mathcal{L} \in EWMON_2^d$. Moreover, it is not hard to verify that the IIM M used in the demonstration of $\mathcal{L} \in EMON_3$ behaves dual monotonically, too.

Finally, $\mathcal{L} \notin CMON_2^d$ can be shown by reducing the halting problem to $\mathcal{L} \in CMON_2^d$. Suppose to the contrary that there are a hypothesis space $\mathcal{G}$ and an IIM M such that M $CMON_2^d$–learns $\mathcal{L}$ with respect to $\mathcal{G}$. By assumption, M, in particular, infers the language L_0 from its text $t = a,\ a^2,\ a^3,\ \ldots$ Consequently, there has to be a least index z such that $M(t_z) = j$ and $L(G_j) = L_0$. Given k and z, simulate M when fed L_{3k+1}'s text $\hat{t} = t_z,\ a^k b,\ a,\ a^2,\ \ldots$ Obviously, M has to change its mind when fed $\hat{t}$. Now, let $r \in \mathbb{N}$ be the least index such that $M(t_z) \neq M(\hat{t}_{z+r})$. It suffices to show that the following recursive predicate ξ may be used to decide whether or not $\varphi_k(k)$ is defined. Let $k \in \mathbb{N}$. We define:

$\xi(k) =$ "Given z and r; test whether or not $\Phi_k(k) \leq z + r + 1$. In case it is, output 1. Otherwise, output 0 and stop."

It remains to show that $\varphi_k(k)$ is undefined if $\xi(k) = 0$. Suppose the converse, i.e., $\xi(k) = 0$ but $\varphi_k(k)$ is defined. Therefore, $\Phi_k(k) = m > z + r + 1$.

Suppose the best case, i.e., M has performed exactly one mind change when fed $\hat{t}_{z+r}$. By definition of $\mathcal{L}$, the segment $\hat{t}_{z+r}$ may serve as an initial segment of a text for each of the pairwise disjoint languages L_{3k+1}, L_{3k+2}, and L_{3k+3}. Since $L_{3k+2} \subset L_{3k+3}$, it is easy to see that M's second guess has to be L_{3k+2}. Otherwise, one can easily define a text for L_{3k+3} on which M needs more than two mind changes in order to infer L_{3k+3}. On the other hand, M dual monotonically infers L_{3k+1} from $\hat{t}$. Therefore, M's second guess has to be different from L_{3k+2}, since otherwise $a^k c^m \in co\text{-}L_0 \cap co\text{-}L_{3k+1}$, but $a^k c^m \notin co\text{-}L_{3k+2} \cap co\text{-}L_{3k+1}$. Consequently, M either fails to work dual monotonically or it needs more than two mind changes, a contradiction. Hence, the predicate ξ solves the halting problem. □

The following subsections are devoted to weak-monotonic inference and dual weak-monotonic learning.

3.5. Weak-Monotonic Learning

Next we consider weak-monotonic learning. Possible relaxations include learning in the limit. However, less is known. First, Theorem 3.7 directly implies $ELIM_1 = EWMON_1$ as well as $LIM_1 = WMON_1$, since every monotonic IIM that learns exactly and class preservingly, respectively, is weak-monotonic, too (cf. Lange and Zeugmann [28]). However, it remains open whether or not $CLIM_1 = CWMON_1$. Nevertheless, we succeeded to obtain results that shed considerable light on the power of learning with at most one mind change.

THEOREM 3.13.

(1) $MON_1 \setminus EWMON \neq \emptyset$,

(2) $ELIM_2 \setminus WMON \neq \emptyset$,

(3) $CMON_1 \setminus WMON \neq \emptyset$.

PROOF. Lange and Zeugmann [29] proved $LIM_1 \setminus EWMON \neq \emptyset$, and recently Lange [26] shows $CLIM_1 \setminus WMON \neq \emptyset$. Combining these results with Theorem 3.7 we directly get Assertion (1) and (3). Finally, for a proof of Assertion (2) we refer the reader to Lange [26]. □

Consequently, relaxing the weak-monotonicity constraint may considerably increase the inference capabilities. However, the latter theorem dealt with indexed families that are themselves not weak-monotonically learnable. Therefore, it is only natural to ask whether or not there are indexed families that can be weak-monotonically inferred within an *a priori* bounded number of mind changes and that are learnable in the limit with less mind changes. The affirmative answer is provided by our next theorem. In particular, we show that unconstrained IIMs may be much more efficient than weak-monotonic machines. In Kinber [25] a similar result concerning the learnability of classes of recursively enumerable languages has been shown. We modify the construction underlying Kinber's proof in order to achieve the following result.

THEOREM 3.14. *For every $n \in \mathbb{N}$, $n \geq 2$, there is an indexed family $\mathcal{L}$ such that*

$\mathcal{L} \in (ELIM_2 \cap CWMON_{n+1}) \setminus CWMON_n$.

PROOF. Assume any $n \geq 2$. Let s_0, s_1, s_2, ... be the lexicographical enumeration of all strings over the underlying terminal alphabet $\Sigma = \{a, b\}$. Given any string $s \in \Sigma^*$ we denote by $\#(s)$ the unique index m such that $s_m = s$. Moreover, assume any standard enumeration M_0, M_1, M_2, ... of all inductive inference machines which accept finite sequences of strings from Σ^* as input. Without loss of generality, we may assume that each of these IIMs, when fed any finite sequence of strings from Σ^*, either outputs a guess or requests the next input (cf. Gold [17]).

Conceptually, the underlying idea is as follows: Given any $j \in \mathbb{N}$, we define a particular family $\mathcal{L}^j$ such that either M_j fails to learn $\mathcal{L}^j$ in a weak-monotonic fashion or M_j needs more than n mind changes. Obviously, in order to show that M_j violates

the weak-monotonicity constraint some *a priori* knowledge about the semantics of M_j's hypotheses is required. In order to provide this knowledge we choose the following approach.

Let φ_0, φ_1, φ_2, ... be any acceptable programming system, and let Φ_0, Φ_1, Φ_2, ... be any fixed associated complexity measure (cf. Section 2). For every $z \in \mathbb{N}$, let $L(\varphi_z) = \{s_m \mid m \in \mathbb{N},\ \varphi_z(m)\downarrow,\ \varphi_z(m) = 1\}$. Then, we use $\mathcal{G} = (L(\varphi_j))_{j\in\mathbb{N}}$, as the universal hypothesis space, i.e., if any of the enumerated IIMs outputs a hypothesis, say k, then we interpret it to mean that the IIM is guessing the language $L(\varphi_k)$. The following lemma guarantees that this approach is successful.

LEMMA 3.4. *Let $\mathcal{L}$ be any indexed family over the terminal alphabet Σ, and $\hat{M}$ be any IIM which CLIM–identifies $\mathcal{L}$ with respect to an admissible hypothesis space $\hat{\mathcal{G}} = (\hat{G}_z)_{z\in\mathbb{N}}$. Then, there exists an IIM M which infers $\mathcal{L}$ with respect to the universal hypothesis space $\mathcal{G}$.*

Recall that membership is uniformly decidable in $\hat{\mathcal{G}}$. For every $z \in \mathbb{N}$, we define the recursive predicate f_z as follows. For all $m \in \mathbb{N}$ we set $f_z(m) = 1$ if and only if $s_m \in L(\hat{G}_z)$. Hence, f_0, f_1, f_2, ... defines an effective enumeration of recursive predicates. Since φ_0, φ_1, φ_2, ... is an acceptable programming system, there exists a recursive compiler c such that $f_z = \varphi_{c(z)}$ for all $z \in \mathbb{N}$. Finally, given this particular compiler c one can easily define an IIM M which learns $\mathcal{L}$ with respect to $\mathcal{G}$. Therefore, let $L \in \mathit{range}(\mathcal{L})$, $t \in \mathit{text}(L)$, and $x \in \mathbb{N}$.

IIM *M*: "Simulate $\hat{M}$ when fed t_x. If $\hat{M}$ outputs a guess, say z, then output $c(z)$. Otherwise, output nothing, and request the next input."

Obviously, M learns $\mathcal{L}$ as required. Note that the above transformation guarantees that any additional constraint $\hat{M}$ satifies, e.g., any of the defined monotonicity constraints or even any requirement on the maximal number of mind changes $\hat{M}$ is allowed to perform, is satisfied by M as well. This proves Lemma 3.4.

Let $j \in \mathbb{N}$. The following procedure $\mathcal{P}$ is used to define the family $\mathcal{L}^j = (L_{\langle j,k\rangle})_{k\in\mathbb{N}}$ in a way such that M_j fails to learn $\mathcal{L}^j$ as required. Thereby, the language $L_{\langle j,0\rangle}$ is specified via its characteristic function $f_{\langle j,0\rangle}$ as a finite subset of $\{a^{j+1}b^m \mid m \in \mathbb{N}\}$. Therefore, it suffices to determine $f_{\langle j,0\rangle}(a^{j+1}b^m)$ for all $m \in \mathbb{N}$. Additionally, all the remaining languages $L_{\langle j,k\rangle}$, $k \geq 1$, either equal the singleton language $\{b\}$ or form a proper subset of $L_{\langle j,0\rangle}$.

***Procedure* $\mathcal{P}$**

Given $j \in \mathbb{N}$, set initially $f_{\langle j,0\rangle}(a^{j+1}) = 1$, $m = 1$, and $W = \{\#(a^{j+1})\}$.

***Stage* 0:** Given m, and W proceed as follows: Test whether or not M_j, when fed $t_{m-1} = \underbrace{a^{j+1}, \ldots, a^{j+1}}_{m-times}$, outputs a guess. In case it does, execute Instruction (A). Otherwise, goto (B).

(A) Set $f_{\langle j,0\rangle}(a^{j+1}b^m) = 1$, and $L_{\langle j,m\rangle} = \{b\}$. Furthermore, set $inp = t_{m-1}$,

$max = m$, $W = W \cup \{\#(a^{j+1}b^{max})\}$, and $m = m + 1$. Goto Stage 1.

(B) Set $f_{\langle j,0\rangle}(a^{j+1}b^m) = 0$, and $L_{\langle j,m\rangle} = \{b\}$. Finally, set $m = m + 1$, and stay in Stage 0.

(* Notice that, in case $\mathcal{P}$ never leaves Stage 0, $L_{\langle j,0\rangle} = \{a^{j+1}\}$ but M_j never outputs a guess when fed $L_{\langle j,0\rangle}$'s text $t = a^{j+1}, a^{j+1}, \ldots$. *)

Stage z $(z \leq n)$**:** (* Recall that n is the number of allowed mind changes. *) Given m, max, inp, and W proceed as follows:

Determine $t_{m-1} = inp$, $\underbrace{a^{j+1}b^{max}, \ldots, a^{j+1}b^{max}}_{(m-max)-times}$. Test whether or not $M_j(inp) \neq M_j(t_{m-1})$. In case it does, goto (A).
Otherwise, determine $p = M_j(inp)$. Test whether or not $\Phi_p(r) \leq m$ for all $r \in W$. If not, follow Instruction (B).
In case it is, test for all $r \in W$ whether or not $\varphi_p(r) = 1$. If yes, then goto (C). Otherwise, execute Instruction (B) as well.

(A) Set $f_{\langle j,0\rangle}(a^{j+1}b^m) = 1$, and $L_{\langle j,m\rangle} = \{b\}$. Additionally, set $inp = t_{m-1}$, $max = m$, $W = W \cup \{\#(a^{j+1}b^m)\}$, and $m = m + 1$. Goto Stage $z + 1$.
(* Notice that inp forms an initial segment of a text for the finite language $L = \{a^{j+1}b^m \mid m \leq max - 1\} \cap L_{\langle j,0\rangle}$ on which M_j, when successively fed it, performs z mind changes. *)

(B) Set $f_{\langle j,0\rangle}(a^{j+1}b^m) = 0$, and $L_{\langle j,m\rangle} = \{b\}$. Do not change max and inp. Furthermore, set $m = m + 1$, and stay in Stage z.

(C) Set $f_{\langle j,0\rangle}(a^{j+1}b^m) = 0$, and $L_{\langle j,m\rangle} = \{b\}$. Finally, set $m = m + 1$, and goto Stage $n + 1$.

(* Notice that, in case $\mathcal{P}$ never leaves Stage z, $L_{\langle j,0\rangle} = inp^+ \cup \{a^{j+1}b^{max}\}$, and M_j, when fed the text $t = inp$, $a^{j+1}b^{max}$, $a^{j+1}b^{max}$, ... for $L_{\langle j,0\rangle}$, converges to $p = M_j(inp)$. *)

Stage $n + 1$**:** Given m, inp, and max proceed as follows: Set $f_{\langle j,0\rangle}(a^{j+1}b^k) = 0$ and set $L_{\langle j,k\rangle} = L_{\langle j,0\rangle} \setminus \{a^{j+1}b^{max}\} = inp^+$ for all $k \geq m$. This finishes the construction of the family $\mathcal{L}^j$.

Since M_j is an IIM and the predicate '$\Phi_i(x) \leq y$' is recursive, it is not hard to verify that $\mathcal{L}^j$ constitutes an indexed family.

Finally, the target indexed family is defined as $\mathcal{L} = (L_{\langle j,k\rangle})_{j,k\in\mathbb{N}}$, i.e., the canonical enumeration of all families $\mathcal{L}^j$.

LEMMA 3.5. $\mathcal{L} \notin CWMON_n$.

Suppose to the contrary that $\mathcal{L} \in CWMON_n$. Hence, there are an IIM $\hat{M}$ and an admissible hypothesis space $\hat{\mathcal{G}}$ such that $\hat{M}$ $CWMON_n$–identifies $\mathcal{L}$ with respect to $\hat{\mathcal{G}}$. By Lemma 3.4 there is an IIM M_j such that M_j weak-monotonically infers $\mathcal{L}$ with

respect to the universal hypothesis space $\mathcal{G}$. Thereby, M_j performs at most n mind changes.

Claim 1. For all $z \leq n$, $\mathcal{P}$ leaves Stage z.

Suppose the converse. First, let $z = 0$. As already mentioned, if $\mathcal{P}$ never leaves Stage 0, then $L_{\langle j,0\rangle} = \{a^{j+1}\}$. Moreover, M_j never outputs a guess, when successively fed the uniquely defined text for $L_{\langle j,0\rangle}$. Hence, M_j fails to learn $L_{\langle j,0\rangle}$, a contradiction. Second, let $0 \leq z < n$. Suppose that $\mathcal{P}$ has successfully finished Stage z, but it never leaves Stage $z+1$. Let *inp* and *max* be given as defined when $\mathcal{P}$ is entering Stage $z+1$. By $\mathcal{P}$'s definition, $L_{\langle j,0\rangle} = inp^+ \cup \{a^{j+1}b^{max}\}$ and M_j, when fed $L_{\langle j,0\rangle}$'s text $t = inp,\ a^{j+1}b^{max},\ a^{j+1}b^{max},\ \ldots$ converges to p. Since M_j learns $L_{\langle j,0\rangle}$, $L(\varphi_p) = L_{\langle j,0\rangle}$. Obviously, this implies $\varphi_p(\#(w)) = 1$ for all $w \in L_{\langle j,0\rangle}$. Consequently, $\mathcal{P}$ sometimes execute Instruction (C). Thus, $\mathcal{P}$ would leave Stage $z+1$, a contradiction. This proves Claim 1.

Furthermore, by assumption, M_j is a weak-monotonic IIM. Hence, M_j never outputs an overgeneralized hypothesis. This implies directly that $\mathcal{P}$ never leaves some Stage z, for some $1 \leq z \leq n$, via Instruction (C). Together with Claim 1 we obtain that $\mathcal{P}$ leaves every Stage via Instruction (A).

Thus, if $\mathcal{P}$ is entering Stage $n+1$, a finite sequence of strings *inp* has been found such that M_j changes its mind n times when fed *inp*. Let $M_j(inp) = p$. By construction, $inp^+ \subset L_{\langle j,0\rangle}$. Now, consider any text t for $L_{\langle j,0\rangle}$ having the prefix *inp*. Since the finite language inp^+ belongs to $\mathcal{L}^j$ as well, we may conclude $L(\varphi_p) \neq L_{\langle j,0\rangle}$. Otherwise, M_j has generated an overgeneralized hypothesis when fed an initial segment of a text for the language inp^+. Consequently, M_j has to perform at least $n+1$ mind changes when inferring $L_{\langle j,0\rangle}$ from t.

Finally, the IIM M_j fails to learn the subfamily $\mathcal{L}^j$ with at most n mind changes. Hence, due to the choice of M_j, in particular $\hat{M}$ does not witness $\mathcal{L}^j \in CWMON_n$. Consequently, $\hat{M}$ does not $CWMON_n$–identify $\mathcal{L}$, a contradiction. This finishes the proof of Lemma 3.5.

LEMMA 3.6. $\mathcal{L} \in CWMON_{n+1}$.

Taking $\mathcal{L}$'s definition into account it is not hard to see that $\mathcal{L}$ contains exclusively finite languages having a cardinality less than or equal to $n+2$. This immediately implies $\mathcal{L} \in CWMON_{n+1}$, since the whole family of all finite languages of cardinality of at most $n+2$ belongs to $CWMON_{n+1}$ (cf. Lange and Zeugmann [29]). Thus, Lemma 3.6 is proved.

LEMMA 3.7. $\mathcal{L} \in ELIM_2$.

Let $L \in range(\mathcal{L})$, $t \in text(L)$, and $x \in \mathbb{N}$. A desired IIM M may work as follows. If $t_x^+ = \{b\}$, then M outputs the least number for the singleton language $\{b\}$ in $\mathcal{L}$. In this case, M clearly learns L. Otherwise, the first string s_0 in t_x determines the unique subfamily $\mathcal{L}^j$ which contains the target language. Then, M starts its work with respect to this subfamily. The only problem M has to handle is to check whether or not there is a proper sublanguage $L_{\langle j,x\rangle}$ of $L_{\langle j,0\rangle}$. However, M has no *a priori* knowledge whether or

not such a language does exist. On the other hand, if there is any such sublanguage, then it equals all the other proper sublanguages of $L_{\langle j,0\rangle}$. Therefore, M may initially output $\langle j,0\rangle$. Subsequently, it uses $x+1$, i.e., the length of its input, to perform a bounded search for the first language $L_{\langle j,y\rangle}$, $1 \leq y \leq x+1$, that does not contain the string b. In case that this search has been successful, M outputs the corresponding hypothesis, say $\langle j,y\rangle$, as long as this guess fulfills $t_x^+ \subseteq L_{\langle j,y\rangle}$. Finally, if M has once verified $t_x^+ \not\subseteq L_{\langle j,y\rangle}$, then it outputs subsequently $\langle j,0\rangle$ on every input t_x. Hence, it performs at most two mind changes.

Moreover, M's correctness immediately follows from the definition of the corresponding subfamily $\mathcal{L}^j$. We omit the details. □

3.6. Dual Weak-Monotonic Learning

Concerning dual weak-monotonic learning much less is known. Nevertheless, our last result points to one major difference between weak-monotonic and dual weak-monotonic inference. In contrast to Theorem 3.14, removing the requirement to learn dual weak-monotonically may never result in an arbitrary large speed-up.

THEOREM 3.15. *For all* $n \in \mathbb{N}^+$: $CLIM_n \subseteq CWMON_{2n}^d$.

PROOF. Let $n \in \mathbb{N}$, and $\mathcal{L} \in CLIM_n$. Assume an IIM M and a hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ such that M $CLIM_n$–identifies $\mathcal{L}$ with respect to $\mathcal{G}$. Without loss of generality, we may assume that M is a consistent IIM (cf. Lange et al. [32]), i.e., if M on input t_x makes an output j_x then $t_x^+ \subseteq L(G_{j_x})$.

Let $\hat{\mathcal{L}} = (\hat{L}_j)_{j\in\mathbb{N}}$ denote any canonical enumeration of all singleton languages over the underlying alphabet and all languages of $\mathcal{L}(\mathcal{G})$. Now we define an IIM $\hat{M}$ that dual weak-monotonically infers $\mathcal{L}$ with respect to $\hat{\mathcal{L}}$. Let $L \in \mathit{range}(\mathcal{L})$, $t \in \mathit{text}(L)$, and $x \in \mathbb{N}$.

IIM $\hat{M}$: "On input t_x do the following: If $x = 0$ or M when fed t_{x-1} does not output any hypothesis, goto (A). Otherwise, goto (B).

(A) Simulate M when fed t_x. If M outputs a guess, say j, then set $FLAG = 1$, output the canonical number of $L(G_j)$ in $\hat{L}$, and request the next input. Otherwise, set $FLAG = 0$, output nothing, and request the next input.

(B) If $FLAG = 1$, test whether or not $M(t_{x-1}) = M(t_x)$. In case it is, repeat $\hat{M}$'s last guess and request the next input. Otherwise, set $FLAG = 0$, output the canonical index of the singleton language t_0^+ in $\hat{\mathcal{L}}$ and request the next input.
If $FLAG = 0$, test whether or not $\mathit{card}(t_x^+) \geq 2$. If not, repeat $\hat{M}$'s last guess and request the next input. Otherwise, let $M(t_x) = j$. Set $FLAG = 1$, output the canonical number of $L(G_j)$ in $\hat{L}$, and request the next input."

First, we show that $\hat{M}$ satisfies the dual weak-monotonical constraint. Let $\hat{M}(t_x) = k \neq \hat{M}(t_{x+1})$. If $FLAG = 1$ when fed t_{x+1}, then $\hat{L}_k = L(G_j)$ where $M(t_x) = j$. Since M

is a consistent IIM, $t_x^+ \subseteq L(G_j)$. Hence, $\hat{M}$ specializes its former guess to the singleton language $L = t_0^+ \subseteq \hat{L}_k$. Otherwise, let $FLAG = 0$ when fed t_{x+1}. By definition, $\hat{L}_k = t_0^+$ and $card(t_x^+) \geq 2$. Thus, $t_x^+ \not\subseteq \hat{L}_k$. Consequently, $\hat{M}$'s mind change is caused by an inconsistency. Hence, $\hat{M}$ is a dual weak-monotonic IIM.

By construction, it is not hard to verify that $card(L) \geq 2$ immediately implies that $\hat{M}$ converges to the same language M converges to. Furthermore, $\hat{M}$ performs at most twice as many mind changes as $\hat{M}$. Finally, let $card(L) = 1$. By construction, $\hat{M}$ learns L with at most one mind change. Consequently, we may conclude that $\hat{M}$ witnesses $\mathcal{L} \in CWMON_{2n}^d$. □

4. Conclusions and Open Problems

In the present paper we have investigated the problem whether or not different monotonicity and dual monotonicity constraints may influence the efficiency of learning algorithms. As we have seen there is no unique answer to this problem. In particular, it turns out that the answer heavily depends on both the (dual) monotonicity demand involved and the set of allowed hypothesis spaces.

For example, if class comprising strong-monotonic and dual strong-monotonic inference is investigated, an arbitrarily large speed-up can be sometimes achieved, even in case that (dual) strong-monotonicity demand is relaxed as weak as possible (cf. Theorems 3.5 and 3.10). However, the situation completely changes in the exact and class preserving strong-monotonic and dual strong-monotonic learning model. If a target indexed family is (dual) strong-monotonically learnable in one of those models then there is definitely *no* IIM *at all* which learns the same family more efficiently (cf. Theorem 3.1 and Proposition 2.3).

In the case of monotonic and dual monotonic inference, respectively, all resulting models share the common property that sometimes an arbitrarily large speed-up may be realized provided that the set of admissible learning algorithms is enlarged to weak-monotonic and dual weak-monotonic IIMs, respectively (cf. Theorems 3.9 and 3.12).

Concerning weak-monotonic learning the resulting picture is less complete. Nevertheless, we have been able to show that the efficiency of weak-monotonic IIMs can arbitrarily increase, if unconstrainted IIMs are admissible (cf. Theorem 3.14). Up to now, this trade off has been detected in the class comprising case, only. Naturally, the question arises whether or not a similar speed-up can be achieved in the exact and class preserving case, too.

Furthermore, we have studied dual weak-monotonic inference. Concerning this approach much less is known. In particular, it remains open what happens in the exact and class preserving case. On the other hand, Theorem 3.15 nicely contrasts Theorem 3.14, i.e., dropping the dual weak-monotonicity constraint can *never* result in an arbitrarily large speed-up. In the best case it is still conceivable that one can halve the number of necessary mind changes if one uses unconstrainted IIMs instead of dual weak-monotonic ones. Nevertheless, it is also imaginable that this relaxation does not

increase the efficiency of learning at all.

Finally, our results show that a relaxation of the corresponding (dual) monotonicity demands may yield a *significant speed-up* of the learning process. Hence, it seems highly desirable to investigate necessary and sufficient conditions $\mathcal{C}_{smon}$ ($\mathcal{C}_{dsmon}$), $\mathcal{C}_{mon}$ ($\mathcal{C}_{dmon}$), and $\mathcal{C}_{wmon}$ ($\mathcal{C}_{dwmon}$) allowing assertions of the following type.

Let LT as well as LT' be any learning type, and let $\mathcal{L} \in LT$. Then one may learn $\mathcal{L}$ more efficiently in the sense of LT' if and only if $\mathcal{C}_{lt'}$ is satisfied but $\mathcal{C}_{lt}$ is not.

Moreover, it would be very interesting to relate possible relaxations of our (dual) monotonicity requirements to problems studied in complexity theory. Recently, such an approach has been undertaken concerning consistent and inconsistent learning resulting in a proof for the superiority of an inconsistent learning algorithm (cf. Wiehagen and Zeugmann [42]). We will see what the future brings concerning these problems.

References

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

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

[3] Angluin, D., and Smith, C.H.: *Inductive inference: theory and methods.* Computing Surveys **15** (1983), 237 – 269.

[4] Angluin, D., and Smith, C.H.: *Formal inductive inference.* In *Encyclopedia of Artificial Intelligence* (St.C. Shapiro, Ed.), Vol. **1** (1987), pages 409 – 418, Wiley-Interscience Publication, New York.

[5] Arikawa, S., Goto, S., Ohsuga, S., and Yokomori, T. (Eds.): *Proceedings 1st International Workshop on Algorithmic Learning Theory.* Japanese Society for Artificial Intelligence, Tokyo, (1990).

[6] Arikawa, S., Maruoka, A., and Sato, T. (Eds.): *Proceedings 2nd International Workshop on Algorithmic Learning Theory.* Japanese Society for Artificial Intelligence, Tokyo, (1991).

[7] Arikawa, S., and Jantke, K.P. (Eds.): *Proceedings 5th International Workshop on Algorithmic Learning Theory* and *Proceedings of the 4th International Workshop on Analogical and Inductive Inference.* Lecture Notes in Artificial Intelligence Vol. **872** (1994), Springer Verlag, Berlin.

[8] Barzdin, Ya.M., and Freivalds, R.V.: *On the prediction of general recursive functions.* Sov. Math. Dokl. **13** (1972), 1224 – 1228.

[9] Barzdin, Ya.M., Kinber, E.B., and Podnieks, K.M.: *Об ускорении синтеза и прогнозирования функций.* In *Теория Алгоритмов и Программ* Vol. **1** (1974), (Ya.M. Barzdin, Ed.) Latvian State University, Riga, pages 117 – 128.

[10] Brewka, G.: *Nonmonotonic Reasoning: Logical Foundations of Commonsense.* Cambridge University Press, Cambridge, (1991).

[11] Case, J.: *The power of vacillation.* In *Proceedings 1st Workshop on Computational Learning Theory*, (D. Haussler and L. Pitt, Eds.), (1988), pages 196 – 205, Morgan Kaufmann Publishers Inc., San Mateo.

[12] Case, J., and Lynes, C.: *Machine inductive inference and language identification.* In *Proceedings 9th International Colloquium on Automata, Languages and Programming*, (M. Nielsen and E.M. Schmidt, Eds.), Lecture Notes in Computer Science Vol. **140** (1982), pages 107 – 115, Springer-Verlag, Berlin.

[13] Case, J., and Smith, C.H.: *Comparison of identification criteria for machine inductive inference.* Theoretical Computer Science **25** (1983), 193 - 220.

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

[15] Fulk, M., and Case, J. (Eds.): *Proceedings 3rd Annual ACM Workshop on Computational Learning Theory.* Morgan Kaufmann Publishers Inc., San Mateo, (1990).

[16] Gasarch, W.I., and Velauthapillai, M.: *Asking questions versus verifiability.* In *Proceedings 3rd International Workshop on Analogical and Inductive Inference*, (K.P. Jantke, Ed.), Lecture Notes in Artificial Intelligence Vol. **642** (1992), pages 197 – 213, Springer-Verlag, Berlin.

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

[18] Haussler, D. (Ed.): *Proceedings 5th Annual ACM Workshop on Computational Learning Theory.* ACM Press, New York, (1992).

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

[20] Jantke, K.P. (Ed.): *Proceedings 2nd International Workshop on Analogical and Inductive Inference.* Lecture Notes in Artificial Intelligence Vol. **397**, (1989), Springer-Verlag, Berlin.

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

[22] Jantke, K.P. (Ed.): *Proceedings 3rd International Workshop on Analogical and Inductive Inference.* Lecture Notes in Artificial Intelligence Vol. **642**, (1992), Springer-Verlag, Berlin.

[23] Kapur, S.: *Monotonic language learning.* In *Proceedings 3rd Workshop on Algorithmic Learning Theory*, (S. Doshita, K. Furukawa, K.P. Jantke and T. Nishida, Eds.), Lecture Notes in Artificial Intelligence Vol. **743** (1992), pages 147 – 158, Springer-Verlag, Berlin.

[24] Kapur, S., and Bilardi, G.: *Language learning without overgeneralization.* In *Proceedings 9th Annual Symposium on Theoretical Aspects of Computer Science*, (A. Finkel and M. Jantzen, Eds.), Lecture Notes in Computer Science Vol. **577** (1992), pages 245 – 256, Springer-Verlag, Berlin.

[25] Kinber, E.B.: *Monotonicity versus efficiency for learning languages from texts.* In *Proceedings 5th Workshop on Algorithmic Learning Theory*, (S. Arikawa and K.P. Jantke, Eds.) Lecture Notes in Artificial Intelligence Vol. **872** (1994), pages 395 – 406, Springer-Verlag, Berlin.

[26] Lange, S.: *The representation of recursive languages and its impact on the efficiency of learning.* In *Proceedings 7th Annual ACM Conference on Computational Learning Theory*, (M. Warmuth, Ed.) pages 256 – 267, (1994), ACM Press, New York.

[27] Lange, S., and Zeugmann, T.: *Types of monotonic language learning and their characterization.* In *Proceedings 5th Annual ACM Workshop on Computational Learning Theory*, (D. Haussler, Ed.), pages 377 – 390, (1992), ACM Press, New York.

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

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

[30] Lange, S., and Zeugmann, T.: *Language learning in dependence on the space of hypotheses.* In *Proceedings 6th Annual ACM Conference on Computational Learning Theory*, pages 127 – 136, (1993), ACM Press, New York.

[31] Lange, S., and Zeugmann, T.: *The learnability of recursive languages in dependence on the space of hypotheses.* Fachbereich Mathematik und Informatik, TH Leipzig, GOSLER–Report 20/93, (July 1993).

[32] Lange, S., Zeugmann, T., and Kapur, S.: *Monotonic and dual-monotonic language learning.* Theoretical Computer Science, to appear.

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

[34] Mukouchi, Y.: *Inductive inference with bounded mind changes.* In *Proceedings 3rd Workshop on Algorithmic Learning Theory*, (S. Doshita, K. Furukawa, K.P. Jantke and T. Nishida, Eds.), Lecture Notes in Artificial Intelligence Vol. **743** (1992), pages 125 – 134, Springer-Verlag, Berlin.

[35] Mukouchi, Y.: *Inductive inference of recursive concepts.* Ph.D. Thesis, RIFIS, Kyushu University 33, RIFIS-TR-CS-82, (March 1994).

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

[37] Rivest, R., Haussler, D., and Warmuth,M. (Ed.): *Proceedings 2nd Annual ACM Workshop on Computational Learning Theory.* Morgan Kaufmann Publishers Inc., San Mateo, (1989).

[38] Shinohara, T.: *Inductive Inference from Positive Data is Powerful.* In *Proceedings 3rd Annual Workshop on Computational Learning Theory*, pages 97 - 110, (1991), Morgan Kaufmann Publishers Inc.

[39] Warmuth, M. (Ed.): *Proceedings 7th Annual ACM Conference on Computational Learning Theory.* ACM Press, New York, (1994).

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

[41] Wiehagen, R., Freivalds, R., and Kinber, B.: *On the power of probabilistic strategies in inductive inference.* Theoretical Computer Science **28** (1984), 111 – 133.

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

[43] Zeugmann, T., and Lange, S.: *A guided tour across the boundaries of learning recursive languages.* Technische Hochschule Leipzig, FB Mathematik und Informatik, GOSLER-Report 26/94, (July 1994).

Received November 30, 1994
Revised January 25, 1995
Communicated by T. Yanagawa