A Guided Tour Across the Boundaries of Learning Recursive Languages

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

Steffen Lange
HTWK Leipzig
FB Mathematik und Informatik
PF 66
04251 Leipzig
steffen@informatik.th-leipzig.de

Abstract

The present paper deals with the learnability of indexed families of uniformly recursive languages *from positive data* as well as from both, *positive and negative data*. We consider the influence of various monotonicity constraints to the learning process, and provide a thorough study concerning the influence of several parameters. In particular, we present examples pointing to typical problems and solutions in the field. Then we provide a *unifying* framework for learning. Furthermore, we survey results concerning learnability in dependence on the hypothesis space, and concerning order independence. Moreover, new results dealing with the efficiency of learning are provided. First, we investigate the power of *iterative* learning algorithms. The second measure of efficiency studied is the number of *mind changes* a learning algorithm is allowed to perform. In this setting we consider the problem whether or not the monotonicity constraints introduced do influence the efficiency of learning algorithms.

The paper mainly emphasis to provide a comprehensive summary of results recently obtained, and of *proof techniques* developed. Finally, throughout our guided tour we discuss the question of what a natural language learning algorithm might look like.

1. Introduction

Humans have the ability to learn and to adapt. During the long evolution of mankind humans have developed the particular ability to acquire their maternal language as well as other languages. Since these abilities have always been considered the hallmark of intelligence, the challenge to design an "intelligent" computer has led to considerable interest in learning in the computer science community. Hence, it would be very nice if we could start our guided tour with a satisfying definition of what learning really is. But our understanding of learning is still too limited. Hence, answering the question what learning is has to be considered to be one of the major goals of algorithmic learning theory (cf. Angluin (1992)). On the other hand, there is

a broad consensus that *induction* constitutes an important feature of learning. The corresponding theory is called *inductive inference*. Inductive inference of formal languages may be characterized as the study of systems that map evidence on a language into hypotheses about it. Of special interest is the investigation of scenarios in which the sequence of hypotheses *stabilizes* to an *accurate* and *finite* description (a grammar) of the target language. If evidence is understood as reasonable information, then all these scenarios model at least a certain aspect of learning. This can be seen as follows. Up to the unknown point of stabilization only finitely many data concerning the target language have been provided. Nevertheless, more information about the language to be learned does not lead to a new, and different hypothesis. Hence, some form of *generalization* must have taken place, i.e., of a grammar that accurately generates the target language. The precise definitions of the concepts "evidence," "stabilization," and "accuracy" go back to Gold (1965, 1967) who introduced the model of learning in the limit. Gold-style formal language learning has been intensively studied (cf., e.g., Angluin and Smith (1983, 1987), Osherson, Stob and Weinstein (1986) and the references therein). 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, Haussler and Warmuth (1989), Fulk and Case (1990), Haussler (1992)), the International Workshops on Algorithmic Learning Theory, ALT (cf., e.g., Arikawa et al. (1990, 1991)) and the workshops on Analogical and Inductive Inference, AII (cf., e.g., Jantke (1989, 1992)).

Most of the work done in the field has been aimed at two goals: the characterization of those collections of languages that can be learned, and to study the impact of several postulates on the behavior of learners to their learning power. Moreover, a considerable amount of interest has been devoted to the learnability of recursively enumerable languages. In this particular setting many interesting and sometimes surprising results have been obtained (cf., e.g., Wiehagen (1978), Case and Lynes (1982), Schäfer-Richter (1984), Case (1988), Jain and Sharma (1989) and Fulk (1990)).

The present paper surveys results that deal with the learnability of *recursive languages*. Looking at potential applications, Angluin (1980a, 1980b) started the systematic study of learning enumerable families of uniformly recursive languages, henceforth called *indexed families*. A sequence $L_0, L_1, L_2, ...$ 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 numbers j. Note that the definition of an indexed families includes both, a description for every language L_j, and a particular enumeration of all the languages. Well-known examples of indexed families are the set of all context sensitive languages in canonical enumeration (cf. Hopcroft and Ullman (1969)) or the set of all pattern languages in canonical enumeration (cf. Angluin (1980a)).

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, we 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.

An algorithmic learner, henceforth called *inductive inference machine* (abbr. IIM), takes as input initial segments of a text (an informant), 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 (informants) for it, then L is said to be *learnable from text* (*learnable from informant*) *in the limit* with respect to the hypothesis space $\mathcal{G}$ (cf. Definition 1).

Having reached that point of precision a question naturally arising is how a "natural" learning algorithm may be designed. A thorough answer to this question, if ever possible, requires a systematic study of the various aspects that might influence the learnability. Our guided tour aims to summarize results obtained in this regard. We continue with some notations that are needed to motivate and to discuss these investigations. The starting point of our studies goes back to different learning strategies that have been discussed controversially in the machine learning community. Clearly, whenever one learns inductively from examples one has to perform a generalization. On the other hand, it is by no means obvious whether one should generalize only as little as necessary or as much as possible. In the first case, the learning algorithm might achieve the learning goal by producing a sequence of better and better generalizations. The second approach might lead to an algorithm that initially outputs a most general hypothesis. Afterwards, the learning algorithm might specialize its actual hypotheses until it eventually reaches a correct guess. Finally, it is plausible to combine the two strategies, i.e., to learn by a suitable interplay between generalization and specialization. There has been an extensive debate in the machine learning community for and against each of these learning modes (cf., e.g., Michalski, Carbonell and Mitchell (1984, 1986) or Kodratoff and Michalski (1990)). Inspired by recent results in non-monotonic reasoning Jantke (1991a, 1991b) proposed several sound formalizations of "generalization." Moreover, he studied the problem to what extend non-monotonic reasoning has to be incorporated into the learning process. Subsequently, Wiehagen (1991) refined Jantke's (1991a) approach, and Kapur (1992) introduced the dual versions of it. This led to the following learning models:

Interpreting generalization and specialization in their strongest sense means that we are forced to produce an augmenting (descending) chain of languages, i.e., $L_i \subseteq L_j$ ($L_i \supseteq L_j$) in case L_j is hypothesized later than L_i (cf. Definitions 5 and 7, Part (A)). The resulting learning types are called *strong-monotonic* and *dual strong-monotonic* learning, respectively.

Subsequently, Wiehagen (1991) refined this definition by restricting "better generalization" to the language L that has to be learned, and required $L_i \cap L \subseteq L_j \cap L$ provided L_j appears later in the sequence of guesses than L_i does (cf. Definition 5 (B)). This means that a new hypothesis is never allowed to reject some string that a previously generated guess already *correctly includes*.

The dual version of the latter requirement directly yields the demand that the learner is never allowed to hypothesize a grammar that can generate a string that a previously guessed hypothesis *correctly excluded* (cf. Definition 7 (B)). Learning devices behaving thus are called *monotonic* and *dual monotonic*, respectively.

Weakening the (dual) strong-monotonicity constraint in the same way as the monotonicity principle of classical logic is generalized to cumulativity (cf., e.g., Brewka (1991)) directly yields *(dual) weak-monotonic* learning, i.e., now the learner is required to behave (dual) strong-monotonic as long as it does not receive data contradicting

its actual hypothesis (cf. Definitions 5 and 7, Part (C)).

Another serious problem one has to deal with when learning from text, is to avoid or to detect *overgeneralizations* (also called the *subset problem*), i.e., hypotheses that describe proper *supersets* of the target language. Several authors proposed the so-called *subset principle* to handle the subset problem (cf., e.g., Berwick (1985), Wexler (1992)). Informally, the subset principle requires the learner to guess the "least" language from the hypothesis space with respect to set inclusion that fits with the data the learner has seen so far. Clearly, each of the monotonicity constraints described above can be regarded as a sound formalization of "least," and therefore, as a realization of the subset principle. But there is another important aspect that we have not touched yet, i.e., the choice of the hypothesis space. Obviously, the hypothesis space must contain at least one description for each target language. Hence, we might be tempted to take the indexed family itself as hypothesis space. And indeed, most authors did (cf. eg. Angluin (1980a, 1980b), Shinohara (1982), Jantke (1991b), Mukouchi (1992)). Moreover, looking at potential applications of a learning system, users of such a system might even be highly interested in getting as hypotheses just the descriptions they proposed. That means they might formulate their learning problems just by specifying a particular indexed family. If an indexed family $\mathcal{L}$ can be learned with respect to $\mathcal{L}$ itself, then we call it *exactly* learnable.

On the other hand, it is only natural to ask whether the requirement to learn exactly may lead to a decrease of the learning power. Results obtained in the setting of PAC-learning impressively show that at least the *efficiency* of learning can be heavily affected if one insists to learn with respect to a particular hypothesis space (cf., e.g., Pitt and Valiant (1988), Blum and Singh (1990)). Similar effects have been observed in Gold-style language learning, too (cf. Lange and Zeugmann (1993b)). Therefore, we also consider the following options to choose a suitable hypothesis space. An indexed family $\mathcal{L}$ is said to be *class preservingly* inferable, if there is a hypothesis space $\mathcal{G} = G_0, G_1, G_2, ...$ such that every grammar G_j generates a language contained in $\mathcal{L}$, and the learning algorithm infers $\mathcal{L}$ with respect to $\mathcal{G}$. Note that this in particular means that *range*$(\mathcal{L})$ and $\{L(G_j) \mid j \in \mathbb{N}\}$ have to coincide ($\mathbb{N}$ is the set of all natural numbers). That means, when dealing with class preserving language learning we are free to choose a possibly *different enumeration* of $\mathcal{L}$ and possibly *different descriptions* for the target languages $L \in \mathcal{L}$. Or in other words, class preserving learning just means that there is at least one suitable hypothesis space having the same range as $\mathcal{L}$ with respect to which $\mathcal{L}$ is inferable.

Finally, we consider *class comprising* learning. In this setting a learning algorithm is allowed to use any hypothesis space $\mathcal{G} = G_0, G_1, ...$ such that every $L \in \mathcal{L}$ possesses a description G_j but $\mathcal{G}$ may additionally contain elements G_k not describing any language from $\mathcal{L}$. Although we might be tempted to exclude class comprising hypothesis spaces, since any grammar not generating a language from $\mathcal{L}$ cannot be correct, this learning model has its peculiarities as we shall see.

Moreover, we study the question what all the described learning models have in *common* and what their *differences* are. As we shall see, *characterizations* are a very useful tool to answer this question (cf. Section 4). In particular, we outline a *unifying framework for learning* from *informant* as well as from *text*.

Next we describe some further requirements that correspond to desirable properties

a "natural" learning algorithm should have. For example, can we require an IIM to be *semantically finite*? An IIM is called *semantically finite* if the hypothesis it converges to is the first correct one in the sequence of all its guesses. Again, this seems to be a very reasonable demand. As we have already mentioned, inductive learning has its peculiarities. There has been a long debate in the philosophy of science for and against induction as a legitimate form of reasoning. As far as learning is concerned, Poper's (1968) *falsification theory*, and his *refutation principle* are of special interest. While finite sequences of data can never prove a hypothesis to be correct, single data can falsify it. Therefore, Popper (1968) considered induction as a legitimate form of drawing conclusions as long as they are built up in such a way that their refutation is possible as long as they are wrong. Adapting that approach to learning we directly arrive at semantical finite learners, since any wrong hypothesis is rejected sometimes, and the sequence of all generated guesses stabilizes to the first correct one. Note that other authors have interpreted Poper's (1968) refutation principle in a different, and much more restrictive way (cf., e.g., Case and Smith (1983), Gasarch and Velauthapillai (1992)).

Furthermore, we deal with the question whether or not the *order* of information presentation does really influence the capabilities of inductive inference machines. Since an IIM is required to learn the target languages from *all* texts (informants), one might conjecture that they just extract the strings provided. While this is true for learning from informant, the situation with respect to text is completely different. This phenomenon has been first observed by Schäfer-Richter (1984), and later independently by Fulk (1990). However, they proved their results in a setting allowing self-referential arguments. Since self-referential arguments are mainly applicable in settings where the membership problem for languages is algorithmically undecidable, it worth to study the problem of order independence in our more realistic setting.

Finally, we consider different aspects dealing with the efficiency of learning. Looking at the definition of learning in the limit, we see that in any learning step an IIM has access to the whole initial segment of a text (informant) it has been fed. Clearly, such a learning mode requires a huge amount of storage. Therefore, we consider *iterative* learning introduced by Wiehagen (1976). An iteratively learning IIM *exclusively* uses its last hypothesis and the next input string to compute its actual guess (cf. Definition 8). Hence that model of learning takes into account the limitation of space in all realistic computations.

The second measure of efficiency we deal with is the number of mind changes an IIM M is allowed to perform. We say that M *changes its mind*, or synonymously, M performs a *mind change* iff two consecutively hypotheses output by M are different (cf. Definition 3). This measure of efficiency has been introduced by Barzdin and Freivalds (1972). Subsequently, various authors used the number of mind changes to characterize the complexity of learning (cf., e.g., Barzdin and Freivalds (1974), Barzdin, Kinber and Podnieks (1974), Case and Smith (1983), Wiehagen, Freivalds and Kinber (1984), Mukouchi (1992, 1994)). Gasarch and Velauthapillai (1992) studied *active learning* in dependence on the number of mind changes.

The paper is organized as follows. Section 2 presents preliminaries, i.e., notations and definitions. In Section 3 we exemplify several basic concepts and ideas of language learning. We continue with characterizations of the learning models introduced

(cf. Section 4). Then, we survey results showing that the learnability of indexed families is sensitive with respect to an appropriate choice of the relevant hypothesis space (cf. Section 5). Fundamental results concerning iterative learning are outlined in Section 6. Subsequently, we present results dealing with the efficiency of learning measured in the number of allowed mind changes (cf. Section 7). A comprehensive summary of recently obtained results dealing with different degrees of order independence is provided in Section 8. Finally, we discuss problems that remain open and outline further questions that might lead to interesting results (cf. Section 9). All references are given in Section 10.

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 ***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 (1978)). Then φ_k is the partial recursive function computed by program k in the programming system. Furthermore, let $k, x \in \mathbb{N}$. If $\varphi_k(x)$ is defined (abbr. $\varphi_k(x)\downarrow$) then we also say that $\varphi_k(x)$ converges; otherwise, $\varphi_k(x)$ diverges (abbr. $\varphi_k(x)\uparrow$). By $\langle\cdot,\cdot\rangle\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$ we denote ***Cantor's pairing function***, i.e., $\langle x, y\rangle = ((x+y)^2+3x+y)/2$ for all $x, y \in \mathbb{N}$. In the sequel we assume familiarity with formal language theory (cf., e.g., Hopcroft and Ullman (1969)). 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. The length of a string $s \in \Sigma^*$ is denoted by $|s|$. Any subset $L \subseteq \Sigma^*$ is called a language. By $co{-}L$ we denote the complement of L, i.e., $co{-}L = \Sigma^* \setminus L$. Let L be a language and $t = s_0, s_1, s_2, ...$ 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. Furthermore, let $i = (s_0, b_0), (s_1, b_1), ...$ be an infinite sequence of elements of $\Sigma^* \times \{+, -\}$ such that $range(i) = \{s_k \mid k \in \mathbb{N}\} = \Sigma^*$, $i^+ = \{s_k \mid (s_k, b_k) = (s_k, +), k \in \mathbb{N}\} = L$ and $i^- = \{s_k \mid (s_k, b_k) = (s_k, -), k \in \mathbb{N}\} = co - L$. Then we refer to i as an ***informant***. If L is classified via an informant then we also say that L is represented by ***positive and negative data***. Let L be a language. By $info(L)$ we denote the set of all informants for L. Moreover, let t, i be a text and an informant, respectively, and let x be a number. Then t_x, i_x denote the initial segment of t and i of length $x+1$, respectively, e.g., $i_2 = (s_0, b_0), (s_1, b_1), (s_2, b_2)$. Let t be a text and let $x \in \mathbb{N}$. Then we define $t_x^+ = \{s_k \mid k \leq x\}$. Furthermore, by i_x^+ and i_x^- we denote the sets $\{s_k \mid (s_k, +) \in i, k \leq x\}$ and $\{s_k \mid (s_k, -) \in i, k \leq x\}$, respectively. Finally, we write $t_x \sqsubseteq t_y$ ($t_x \sqsubset t_y$), iff t_x is a (proper) prefix of t_y.

Next we introduce the notion of the ***canonical text*** that will be very helpful in proving several theorems. Let L be any non-empty recursive language, and let $s_0, s_1, ...$ be the lexicographically ordered text of Σ^*. Test sequentially whether $s_z \in L$, for $z = 0, 1, 2, ...$, 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, $x \geq 0$,

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

Following Angluin (1980b), we restrict ourselves to deal exclusively with indexed families of uniformly recursive languages defined as follows: A sequence $L_0, L_1, L_2, ...$ is said to be an ***indexed family*** $\mathcal{L}$ of uniformly recursive languages provided all L_j are non-empty and there is a recursive function f such that for all numbers j and all strings $s \in \Sigma^*$ we have

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

In the following we refer to indexed families of uniformly recursive languages as indexed families for short. Moreover, we sometimes denote an indexed family and its range by the same symbol $\mathcal{L}$. The meaning will be clear from the context.

As in Gold (1967), we define an ***inductive inference machine*** (abbr. IIM) to be an algorithmic device which works as follows: The IIM takes as its input larger and larger initial segments of a text t (or an informant i) and it either requests the next input, or it first outputs a hypothesis, i.e., a number encoding a certain computer program, and then it requests the next input (cf., e.g., Angluin (1980b)).

At this point we have to clarify what space of hypotheses we should choose, thereby also specifying the goal of the learning process. Gold (1967) and Wiehagen (1977) pointed out that there is a difference in what can be inferred depending on whether we want to synthesize in the limit grammars (i.e., procedures generating languages) or decision procedures, i.e., programs of characteristic functions. Case and Lynes (1982) investigated this phenomenon in detail. As it turns out, IIMs synthesizing grammars can be more powerful than those ones which are requested to output decision procedures. However, in the context of identification of indexed families, both concepts are of equal power. Nevertheless, we decided to require the IIMs to output grammars. This decision has been caused by the fact that there is a big difference between the possible monotonicity requirements. A straightforward adaptation of the approaches made in inductive inference of recursive functions directly yields analogous requirements with respect to the corresponding characteristic functions of the languages to be inferred. On the other hand, it is only natural to interpret monotonicity with respect to the language to be learned, i.e., to require containment of languages as described in the introduction. It turns out that the latter approach considerably increases the power of all types of monotonic and dual monotonic language learning. Furthermore, 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 $\mathcal{G} = G_0, G_1, G_2, ...$ over the terminal alphabet Σ satisfying $\mathcal{L} \subseteq \{L(G_j) \mid j \in \mathbb{N}\}$. Moreover, we require that membership in $L(G_j)$ is uniformly decidable for all $j \in \mathbb{N}$ and all strings $s \in \Sigma^*$. As it turns out, it is sometimes very important to choose the space of hypotheses appropriately in order to achieve the desired learning goal (cf., e.g., Section 5 and 8). 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. Then we set $\mathcal{L}(\mathcal{G}) = \{L(G_j) \mid j \in \mathbb{N}\}$. Note that $\mathcal{L}(\mathcal{G})$ constitutes itself an indexed family for all hypothesis spaces $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$.

Let σ be a text or informant, respectively, and $x \in \mathbb{N}$. Then we use $M(\sigma_x)$ to denote the last hypothesis produced by M when successively fed σ_x. The sequence $(M(\sigma_x))_{x \in \mathbb{N}}$ is said to ***converge in the limit*** to the number j if and only if either $(M(\sigma_x))_{x \in \mathbb{N}}$ is infinite and all but finitely many terms of it are equal to j, or

$(M(\sigma_x))_{x\in\mathbb{N}}$ is non-empty and finite, and its last term is j. Now we are ready to define learning in the limit.

Definition 1. (Gold, 1967) *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–TXT [CLIM–INF]–identifies L from text [informant] with respect to $\mathcal{G}$*** *iff for every text t [informant i] for L, there exists a $j \in \mathbb{N}$ such that the sequence $(M(t_x))_{x\in\mathbb{N}}$ $[(M(i_x))_{x\in\mathbb{N}}]$ converges in the limit to j and $L = L(G_j)$.*

Furthermore, $M\ CLIM{-}TXT$ $[CLIM{-}INF\ ]$–identifies $\mathcal{L}$ with respect to $\mathcal{G}$ iff, for each $L \in \mathcal{L}$, $M\ CLIM{-}TXT$ $[CLIM{-}INF\ ]$–identifies L with respect to $\mathcal{G}$.

Finally, let $CLIM{-}TXT$ $[CLIM{-}INF\ ]$ 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 $CLIM{-}TXT$ $[CLIM{-}INF\ ]$–identifies $\mathcal{L}$ with respect to $\mathcal{G}$.

Since, by the definition of convergence, only finitely many data of L were seen by the IIM upto the (unknown) point of convergence, whenever an IIM identifies the language L, some form of learning must have taken place. For this reason, hereinafter the terms *infer*, *learn*, and *identify* are used interchangeably.

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.

Note that, in general, it is not decidable whether or not M has already inferred L. With the next definition, we consider a special case where it has to be decidable whether or not an IIM has successfully finished the learning task.

Definition 2. (Gold, 1967; Trakhtenbrot and Barzdin, 1970) *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 CFIN–TXT [CFIN–INF]–identifies L from text [informant] with respect to $\mathcal{G}$*** *iff for every text t [informant i] for L, there exists a $j \in \mathbb{N}$ such that M, when successively fed t [i], outputs the single hypothesis j, $L = L(G_j)$, and stops thereafter.*

Furthermore, $M\ CFIN{-}TXT$ [$CFIN{-}INF\ ]$–identifies $\mathcal{L}$ with respect to $\mathcal{G}$ iff, for each $L \in \mathcal{L}$, $M\ CFIN{-}TXT$ [$CFIN{-}INF\ ]$–identifies L with respect to $\mathcal{G}$.

The resulting learning type is denoted by $CFIN{-}TXT$ $[CFIN{-}INF\ ]$.

The next definition shows a natural way of weakening the requirement of finite identification. Here, the number of mind changes which an IIM M may perform when inferring a target language is bounded by a number *a priori* fixed. When dealing with mind changes it is technically much more convenient to require the IIMs to behave as follows. Let t be any text (i be any informant), and $x \in \mathbb{N}$. If M on t_x (i_x) outputs for the first time a guess, then it has to output at any subsequent step a hypothesis. It is easy to see that any IIM M may be straightforwardly converted into an IIM $\hat{M}$

behaving as required such that both machines produce the same sequence of mind changes.

Definition 3. (Barzdin and Freivalds, 1974) *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, $k \in \mathbb{N} \cup \{*\}$.* ***An IIM $\mathbf{CLIM_k}$–TXT [$\mathbf{CLIM_k}$–INF]–identifies $\mathcal{L}$ from text [informant] with respect to $\mathcal{G}$*** *iff*

(1) *M $CLIM-TXT$ $[CLIM-INF]$–identifies L from text [informant] with respect to $\mathcal{G}$,*

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

Moreover, M $CLIM_k-TXT$ [$CLIM_k-INF$]–identifies $\mathcal{L}$ with respect to $\mathcal{G}$ iff, for each $L \in \mathcal{L}$, M $CLIM_k-TXT$ [$CLIM_k-INF$]–identifies L with respect to $\mathcal{G}$.

$CLIM_k-TXT$ and $CLIM_k-INF$ are defined in the same way as above.

Obviously, $\lambda FIN-TXT = \lambda LIM_0-TXT$ and $\lambda FIN-INF = \lambda LIM_0-INF$ for all $\lambda \in \{E, \varepsilon, C\}$. Moreover, it is easy to see that $\lambda LIM_*-TXT = \lambda LIM-TXT$ as well as $\lambda LIM_*-INF = \lambda LIM-INF$ for all $\lambda \in \{E, \varepsilon, C\}$.

Next, we want to formally define strong-monotonic, monotonic and weak-monotonic inference. But before doing this, we first define *consistent* identification. Consistent learning devices have been introduced by Barzdin (1974). Intuitively, consistency means that the IIM has to correctly reflect the information it has already been fed.

Definition 4. (Barzdin, 1974) *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 CCONS–TXT [CCONS–INF]–identifies $\mathcal{L}$ from text [informant] with respect to $\mathcal{G}$*** *iff*

(1) *M $CLIM-TXT$ $[CLIM-INF]$–identifies L from text [informant] with respect to $\mathcal{G}$,*

(2) *for every text t [informant i] for L the following condition is satisfied: whenever M on t_x [on i_x] produces a hypothesis j_x, then $t_x^+ \subseteq L(G_{j_x})$ [$i_x^+ \subseteq L(G_{j_x})$ and $i_x^- \subseteq co-L(G_{j_x})$].*

Moreover, M $CCONS-TXT$ [$CCONS-INF$]–identifies $\mathcal{L}$ with respect to $\mathcal{G}$ iff, for each $L \in \mathcal{L}$, M $CCONS-TXT$ [$CCONS-INF$]–identifies L with respect to $\mathcal{G}$.

$CCONS-TXT$ and $CCONS-INF$ are analogously defined as above.

Now we are ready to formally define the three types of monotonic language learning introduced in Section 1.

Definition 5. (Jantke, 1991a; Wiehagen, 1991) *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 [informant] with respect to $\mathcal{G}$***

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

(B) ***monotonically***

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

iff

M $CLIM-TXT$ [$CLIM-INF$]–identifies L with respect to $\mathcal{G}$ and for every text t [informant i] 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} [i_x and i_{x+k}], 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* $L(G_{j_x}) \subseteq L(G_{j_{x+k}})$ [*if* $i^+_{x+k} \subseteq L(G_{j_x})$ *and* $i^-_{x+k} \subseteq co-L(G_{j_x})$, *then* $L(G_{j_x}) \subseteq L(G_{j_{x+k}})$].

In particular, requirement (C) means that M behaves strong-monotonically as long as its guess j_x is consistent with *all* the data fed to M both before and after M has output j_x.

We denote by $CSMON-TXT$, $CSMON-INF$, $CMON-TXT$, $CMON-INF$, $CWMON-TXT$, $CWMON-INF$ 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 or informant with respect to $\mathcal{G}$, respectively.

With the following figure we summarize the known results concerning monotonic language learning (cf. Lange and Zeugmann (1992, 1993a)). We restrict ourselves to the class preserving case, since this case already reflects the characteristic relations between the monotonic learning models defined above. Each learning type is represented as a vertex in a directed graph. A directed edge from vertex A to vertex B indicates that A is a proper subset of B, a bidirectional edge represents $A = B$, and no edge between vertices not connected by a directed path implies that A and B are incomparable.

Monotonic Learning from Text versus Monotonic Learning from Informant

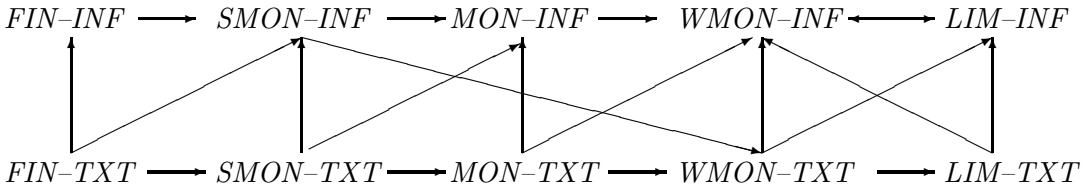

Figure 1

Next, we define *conservative* IIMs. Intuitively speaking, conservative IIMs maintain their actual hypothesis at least as long as they have not seen data contradicting it. Hence, whenever a conservative IIM performs a mind change it is because it has perceived clear inconsistency between its guess and the input.

Definition 6. (Angluin, 1980b) *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 CCONSERVATIVE–TXT [CCONSERVATIVE–INF]–identifies L from text [informant] with respect to $\mathcal{G}$*** *iff*

(1) *M CLIM−TXT [CLIM−INF]–identifies L with respect to $\mathcal{G}$,*

(2) *for every text t [informant i] for L the following condition is satisfied: whenever M on input t_x [on i_x] makes the guess j_x and then makes the guess $j_{x+k} \neq j_x$ at some subsequent step, then $L(G_{j_x})$ must fail to contain some string from t^+_{x+k} [$L(G_{j_x})$ must either fail to contain some string $s \in i^+_{x+k}$ or it generates some string $s \in i^-_{x+k}$].*

Finally, M CCONSERVATIVE–TXT [CCONSERVATIVE–INF]–identifies $\mathcal{L}$ with respect to $\mathcal{G}$ if and only if, for each $L \in \mathcal{L}$, M CCONSERVATIVE–TXT [CCONSERVATIVE–INF]–identifies L with respect to $\mathcal{G}$.

The collection of sets *CCONSERVATIVE–TXT* and *CCONSERVATIVE–INF* are defined in a manner analogous to that above.

Note that $\lambda WMON\text{–}TXT = \lambda CONSERVATIVE\text{–}TXT$ as well as $\lambda WMON\text{–}INF = \lambda CONSERVATIVE\text{–}INF$ for all $\lambda \in \{C, \varepsilon, E\}$ (cf. Lange and Zeugmann (1993a)). Hence, looking at Figure 1 we may conclude that conservative IIMs are less powerful than unrestricted IIMs, in case one deals with the inferability of indexed families. Note that the latter assertion is not true if one deals with the learnability of arbitrary *recursively enumerable* languages (cf. Osherson, Stob and Weinstein (1986), pp. 75).

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

Definition 7. (Kapur, 1992) *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 a language L from text [informant] with respect to $\mathcal{G}$***

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

(B) ***dual monotonically***

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

iff

M CLIM−TXT [CLIM−INF]–identifies L with respect to $\mathcal{G}$ and for any text t [informant i] 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} [i_x and i_{x+k}], for some $k \geq 1, k \in \mathbb{N}$, the following conditions are satisfied:

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

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

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

By $CSMON^d\text{–}TXT$, $CSMON^d\text{–}INF$, $CMON^d\text{–}TXT$, $CMON^d\text{–}INF$, $CWMON^d\text{–}TXT$, and $CWMON^d - INF$ 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 and informant with respect to $\mathcal{G}$, respectively.

The next figure shows the relations between the defined modes of class preserving dual monotonic inference (cf. Lange, Zeugmann and Kapur (1992), and Lange and Zeugmann (1994)). The semantics of Figure 2 is analogous to that of Figure 1. On comparing with Figure 1, the similarities as well as the differences between the various types of monotonic and dual monotonic inference are clearly illustrated.

Dual Monotonic Learning from Text versus Dual Monotonic Inference from Informant

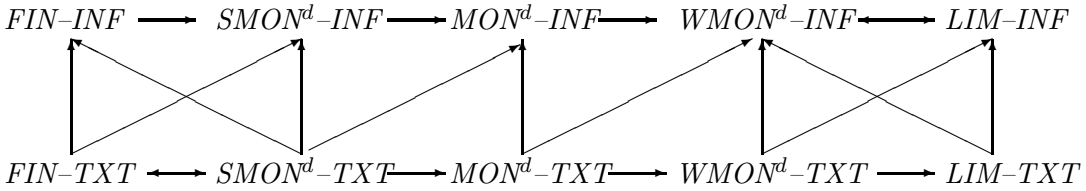

Figure 2

Note that the notions of monotonicity and of dual monotonicity are truly duals of *each other*.

Finally, we define iterative IIMs. An iterative IIM is only allowed to use its last guess and the next string of the text and informant, respectively, of the language it is supposed to learn. Conceptionally, an iterative IIM M defines a sequence $(M_n)_{n\in N}$ of machines each of which takes as its input the output of its predecessor. Hence, the IIM M has always to produce a hypothesis.

Definition 8. (Wiehagen, 1976) *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 CIT–TXT [CIT–INF]–identifies L from text [informant] with respect to $\mathcal{G}$*** *iff for every text $t = (s_j)_{j\in\mathbb{N}}$ [informant $i = ((w_j, b_j))_{j\in\mathbb{N}}$] the following conditions are satisfied:*

(1) *for all $n \in \mathbb{N}$, $M_n(t)$ [$M_n(i)$] is defined, where $M_0(t) =_{df} M(s_0)$ [$M_0(i) =_{df} M((w_0, b_0))$] and for all $n \geq 0$: $M_{n+1}(t) =_{df} M(M_n(t), s_{n+1})$ [$M_{n+1}(i) =_{df} M(M_n(i), (w_{n+1}, b_{n+1}))$],*

(2) the sequence $(M_n(t))_{n\in\mathbb{N}}$ [$(M_n(i))_{n\in\mathbb{N}}$] converges in the limit to a number j such that $L = L(G_j)$.

Finally, M $CIT-TXT$ [$CIT-INF$]–identifies $\mathcal{L}$ with respect to $\mathcal{G}$ iff, for each $L \in \mathcal{L}$, M $CIT-TXT$ [$CIT-INF$]–identifies L with respect to $\mathcal{G}$.

The resulting identification types $CIT-TXT$ and $CIT-INF$ are analogously defined as above.

The combination of iterative and monotonic inference is denoted by $\lambda MON-IT-TXT$ ($\lambda MON-IT-INF$), where $\lambda \in \{S, W, \varepsilon\}$.

The next section starts our guided tour across the boundaries of learning recursive languages. We begin with several examples pointing to typical problems, ideas, and solutions in the field.

3. Examples

One of the first discovered learning algorithms has been *identification by enumeration* (cf. Solomonoff (1964), Gold (1965)). Nowadays, this learning algorithm is usually referred to as Gold's *identification by enumeration principle*. The main idea behind this algorithm is as follows. Choose a suitable enumeration of all the objects to be learned. Then, after having seen the data $d_0, ..., d_x$, search for the first enumerated object that is consistent with the data read so far. However, at first glance it seemed that this learning procedure has a severely restricted domain of applicability, at least as long as its effective computability is required. Clearly, as it stands, its effective computability can only be assured as long as the corresponding consistency problem remains effectively decidable. Moreover, it turned out that consistency itself constitutes a severe restriction of learnability (cf. e.g. Wiehagen and Zeugmann (1994), and the references therein). Nevertheless, suitable modifications of the identification by enumeration principle have been discovered that turned out to be very powerful. Nowadays, for the setting of inductive inference of recursive functions the following thesis is widely accepted (cf. Wiehagen (1991), pp. 184):

"Any class of recursive functions which is identifiable at all can always be identified by an ***enumeratively working*** *learning device. Moreover, the identification can always be realized with respect to a suitable non-standard (i.e., non-Gödel) numbering of the target class of functions."*

Thus, the question arises whether or not enumeratively working IIMs are of the same importance in language learning, too.

We start our guided tour with a series of examples pointing to major problems in answering the latter question. Furthermore, we exemplify fundamental ideas that have been developed to handle the arising difficulties. Thereby we restrict ourselves to the learnability of indexed families. Let us start with the easiest case, i.e., with the learnability of indexed families from positive and negative data. Inference from informant may be understood, at least conceptually, as inductive inference of enumerable classes of recursive predicates. Therefore, it is easy to see that Gold's (1967) *identification by enumeration principle* serves as a universal learning method. Moreover, we may even use any hypothesis space comprising all the target languages. In particular, successful inference can be always achieved, if we choose the target indexed family itself as the underlying hypothesis space. For the sake of completeness, let us continue with the definition of an IIM that realizes the identification by enumeration principle.

Let $\mathcal{L}$ be any indexed family, $L \in \mathcal{L}$, let $i \in \mathit{info}(L)$, and $x \in \mathbb{N}$. The wanted IIM M works as follows. When fed i_x it searches for the least index j satisfying $i_x^+ \subseteq L_j$ and $i_x^- \cap L_j = \emptyset$, i.e., the first enumerated language that is consistent with all data read so far. Then, M outputs the hypothesis j.

Some remarks are mandatory. Since membership is uniformly decidable for all languages enumerated in $\mathcal{L}$, the consistency test can be effectively performed. Moreover, since $i \in \mathit{info}(L)$ and $L \in \mathcal{L}$, the described search has to terminate. Hence, M is indeed an IIM. Furthermore, M converges to the least number j satisfying $L = L_j$, and performs at most j mind changes. But still, this is not the whole story. The described IIM M possesses some further advantages that we are going to describe. First, any mind change performed by M is justified by a "provable misclassification" of its

previous guess. Therefore, M will never reject a guess that is correct for the language to be learned. Hence, M is *semantical finite*. Second, M is *set-driven*, too, i.e., its output exclusively depends on the range of its input. Next, identification by enumeration is the most efficient learning method with respect to learning time, i.e., the first time such that M outputs a correct guess that will be repeated in every subsequent learning step (cf. Gold (1967)). More precisely, Gold proved that there is no IIM $\hat{M}$ inferring $\mathcal{L}$ which is *uniformly faster* than the IIM M described above with respect to learning time. Hence, in the setting of learning indexed families identification by enumeration is particularly tailored for learning from informant.

However, the situation remarkably changes when learning from positive data is concerned. There are several reasons for that phenomenon. The first one is a topological one, and has been discovered by Gold (1967). In particular, he proved the following theorem.

Proposition 1. (Gold, 1967) *Let $\mathcal{L}$ be any class of languages containing all finite languages and at least one infinite language. Then $\mathcal{L} \notin CLIM{-}TXT$.*

Note that Proposition 1 remains true, even in case one restricts itself to learning from *recursive* text (cf. Gold (1967)). Moreover, a closer look to Gold's proof directly implies that even quite simple indexed families are not identifiable from text as our first example shows.

Example 1. Nonlearnability of simple indexed families

Consider the following indexed family $\mathcal{L} = (L_j)_{j \in \mathbb{N}}$ where $L_0 = \{a\}^+$ and $L_j = \{a^m \mid 1 \leq m \leq j\}$ for all $j \in \mathbb{N}^+$. We show that $\mathcal{L}$ is not learnable in the limit from positive data. Suppose the converse, i.e., there is an IIM M which witnesses $\mathcal{L} \in ELIM{-}TXT$. Then, M in particular has to identify the language L_1 on its uniquely defined text t. Hence, there exists an $x \in \mathbb{N}$ such that $M(t_x) = 1$. Obviously, it is possible to extend t_x in order to obtain a text for the infinite language L_0. Namely, we may choose the text $t_x \cdot \hat{t}$ where $\hat{t}$ is the lexicographically ordered text of L_0. Since M has to infer L_0 from this text, too, M is forced to change its mind to the hypothesis 0. Therefore, there is a $y \in \mathbb{N}^+$ such that $M(t_x \cdot \hat{t}_y) = 0$. But now we may conclude that $t_x \cdot \hat{t}_y$ is an initial segment of a text $\tilde{t}$ for the finite language L_y. Consequently, M has to perform one more mind change when successively fed $\tilde{t}$. By iterating this idea one may effectively construct a text for the infinite language L_0 on which M has to change its mind infinitely often, a contradiction. Hence $\mathcal{L} \notin ELIM{-}TXT$. As we shall see later (cf. Section 5, Theorem 11), this result implies that $\mathcal{L}$ is not learnable at all in the limit, i.e., $\mathcal{L} \notin CLIM{-}TXT$. $\diamond$

This negative result is mainly caused by the problem that both finite and infinite languages have to be simultaneously handled. Moreover, Proposition 1 as well as our example do not only point to a weakness of identification by enumeration but to a serious weakness of learning from text. Hence, it is still imaginable that identification by enumeration remains a universal learning method provided the target indexed family is learnable in the limit from positive data. Our next example shows that the situation is much more subtle than one might expect.

Example 2. The weakness of identification by enumeration

Consider the following indexed family $\mathcal{L} = (L_j)_{j \in \mathbb{N}}$ where $L_0 = \{a\}^+$ and $L_j = \{a^j\}$ for all $j \in \mathbb{N}^+$. An IIM $\hat{M}$ which conservatively infers $\mathcal{L}$ may be designed as follows.

As long as a text for a singleton language, say L_j, is presented, $\hat{M}$ outputs just the guess j. If at least two different strings appear, $\hat{M}$ changes its mind to its final guess 0. Thus, on the one hand, $\mathcal{L} \in ELIM{-}TXT$.

But on the other hand, any IIM M realizing the identification by enumeration principle cannot infer $\mathcal{L}$ from positive data. To see this, suppose the converse, i.e., there is such an IIM M inferring $\mathcal{L}$ with respect to any class comprising hypothesis space $\mathcal{G}$. Hence, there is a least index $z \in \mathbb{N}$ such that $L_0 = L(G_z)$. Therefore, there has to be a singleton language L satisfying $L(G_k) \neq L$ for all $k \leq z$. Since $L \subseteq L(G_z)$, M will never output a hypothesis $j > z$ when fed a text for L. Thus, M fails to infer the singleton language L. $\diamond$

The indexed family $\mathcal{L}$ from the above example contains a language L as well as infinitely many proper sublanguages of L. A constellation like this always implies that identification by enumeration fails to learn the corresponding target indexed family from text. As mentioned above, when learning from informant is considered identification by enumeration is insensible to the choice of the hypothesis space as long as learnability at all is concerned. Our next example shows that this is no longer the case when learning from positive data is considered.

Example 3. Sensibility of identification by enumeration with respect to the hypothesis space

Consider the indexed family $\mathcal{L} = (L_j)_{j \in \mathbb{N}}$ where $L_0 = \{a\}^+$ and $L_j = \{a\}$ for all $j \in \mathbb{N}^+$. Again, identification by enumeration fails when $\mathcal{L}$ is selected as hypothesis space. On the other hand, identification by enumeration yields successful inference, if a hypothesis space $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ is chosen such that $L(G_0) = \{a\}$ and $L(G_j) = \{a\}^+$ for all $j \in \mathbb{N}^+$. $\diamond$

Taking the above examples into consideration, the question arises whether or not identification by enumeration may be suitably modified for language learning from positive data. The crucial point is how overgeneralization may be avoided or, in case overgeneralization is inevitable, how the resulting problems may be handled. Furthermore, it would be interesting to know if all, or at least some, of the useful properties of identification by enumeration can be maintained. After Gold's (1967) pioneering paper these problems faced more than a decade of decline. Proposition 1 had been misinterpreted. Namely, many authors concluded that there is no interesting class of languages at all that can be learned from positive data, and hence, there is no need to study the question mentioned above. The breakthrough has been provided by Angluin (1980a) who showed that there are very interesting languages that can be inferred from text, for example the class of all pattern languages. Subsequently, the learnability of pattern languages has been intensively studied within different learning models (cf. e.g. Shinohara (1982), Kearns and Pitt (1989), Lange and Wiehagen (1991)). Besides that, pattern languages form the basis of applications in different fields, e.g. in the "intelligent" text processing system EBE (cf. Nix (1983)) or in a classification system for transmembrane proteins (cf. Arikawa et. al (1992)). So let us have a closer look to them.

Example 4. The pattern languages

Let $\Sigma = \{a, b, ..\}$ be any non-empty finite alphabet containing at least two letters. Furthermore, let $X = \{x_0, x_1, x_2, ...\}$ be an infinite set of variables such that $\Sigma \cap X = \emptyset$.

Patterns are non-empty strings from $\Sigma \cup X$, e.g., ab, ax_1ccc, $bx_1x_1cx_2x_2$ are patterns. If p is a pattern, then $L(p)$, the language generated by pattern p, is the set of strings which can be obtained by substituting non-null strings $s_i \in \Sigma^*$ for each occurrence of the variable x_i in the pattern p. Thus $aabbb$ is generable from pattern ax_1x_2b, while $aabba$ is not. From a practical point of view it is highly desirable to choose the hypothesis space as small as possible. For that purpose we use the canonical form of patterns (cf. Angluin (1980a)). A pattern p is in *canonical form* provided that if k is the number of variables in p, then the variables occurring in p are precisely $x_0, ..., x_{k-1}$. Moreover, for every j with $0 \leq j < k-1$, the leftmost occurrence of x_j in p is to the left of the leftmost occurrence of x_{j+1} in p. If a pattern p is in canonical form then we refer to p as a canonical pattern. Let $Patc$ denote the set of all canonical patterns. Clearly, for every pattern p there exists a unique $q \in Patc$ such that $L(p) = L(q)$. Finally, choose any repetition free effective enumeration $p_0, p_1, ...$ of $Patc$ and define $PAT = (L(p_j))_{j \in \mathbb{N}}$. Then PAT establishes an indexed family (cf. Angluin (1980a)).

In the sequel, we discuss the learnability of the family PAT from different perspectives. Thereby, we are mainly interested in principal solutions. Technicalities are suppressed as much as possible.

First, we summarize some characteristic features of pattern languages. From the above definition it immediately follows that given any two patterns p and q it is decidable whether or not $L(p) = L(q)$. Furthermore, let $T_p = \{w \mid w \in L(p),\ |w| = |p|\}$ for every pattern $p \in Patc$. Then, every pattern language $L(p)$ is uniquely characterized by its finite subset T_p as the following proposition shows.

Proposition 2. (Angluin, 1980a) *Let p, q be any patterns. Then we have:*

(1) $L(p) = L(q)$ *iff* $T_p \subseteq L(q)$ *and* $T_q \subseteq L(p)$,

(2) $T_p \subseteq L(q)$ *implies* $\neg\, L(q) \subset L(p)$.

Let S denote any finite set of strings. A pattern p is said to be ***descriptive for S*** provided that $S \subseteq L(p)$ and there does not exists any pattern q satisfying $S \subseteq L(q) \subset L(p)$. Therefore, any IIM M which exclusively outputs descriptive patterns as hypotheses realizes the subset principle, i.e., it never generates an overgeneralized hypothesis. The sets T_p, henceforth called *tell-tale sets*, can be effectively computed, if p is given. Hence, tell-tale sets may guide a learning device to avoid the problem of overgeneralization.

Based on this idea, the following IIM conservatively infers PAT with respect to the hypothesis space PAT. Assume that any initial segment t_x of any text t for any pattern language $L(q)$ is presented. Initially, M searches for all indices $j \leq x$ such that $t_x^+ \subseteq L(p_j)$. Among these candidate hypotheses, M chooses the least j satisfying $T_{p_j} \subseteq t_x^+$, if such an index exists. Because of Assertion (2), p_j is descriptive for the set T_{p_j}. Since $T_{p_j} \subseteq t_x^+ \subseteq L(p_j)$, p_j is descriptive for the set t_x^+, too. Thus, M has seen enough evidence to output just the hypothesis j. M does not change the hypothesis j as long as it is consistent. If j turns out to be inconsistent then M again starts the same search as explained above. Now, taking Proposition 2 into account, it is easy to see that M converges on t. Moreover, since M never outputs an overgeneralized hypothesis, M learns $L(q)$ from text. Thus, we have the following theorem.

Theorem 1. $PAT \in ECONSERVATIVE - TXT$.

The IIM M explained above works enumeratively, too. But, M's search within the hypothesis space does not only aim to find any consistent hypothesis. Instead, M does not output any hypothesis until it has collected enough evidence. It measures the evidence obtained with respect to the relevant tell-tale sets. If at least one tell-tale set is completely contained in the range of its input M creates a suitable subspace of candidate hypotheses, i.e., the set of all those patterns p satisfying $T_p \subseteq t_x^+$. Then it searches the first consistent hypothesis, say p, and outputs it. If p is not a correct guess, Assertion (2) of Proposition 2 guarantees that data contradicting it have to appear.

As we have seen, there is some hope to suitably modify identification by enumeration for learning from positive data. Moreover, for the particular case of pattern languages the tell-tale sets are the main ingredient to solve the subset problem. Consequently, it is only natural to ask whether or not the approach outlined above may be generalized. This is indeed the case as we shall see below. But before dealing with possible generalizations we provide some more information concerning the learnability of pattern languages. First, we ask whether Theorem 1 may be strengthened, for example to $PAT \in SMON-TXT$. Recently, it has been shown that the inclusion problem for pattern languages is undecidable. This implies the following result (cf. Zeugmann, Lange and Kapur (1995) for a detailed discussion).

Theorem 2. $PAT \notin MON-TXT$.

Hence, Theorem 1 cannot be improved with respect to class preserving learning. On the other hand, the pattern languages possess another favorable property, namely they are iteratively learnable (cf. Lange and Wiehagen (1991)).

Theorem 3. $PAT \in EIT-TXT$.

The basic idea may be described as follows. Ignore for a moment that the desired IIM has to be iterative. As we have already seen, every pattern language is uniquely characterized by the set of all its minimal strings. This observation is the main ingredient to the following IIM M. At each learning step, M first determines the set S of all minimal strings it has been fed so far. All other strings will be simply ignored. Then, M generates a pattern p_j that is descriptive for S. Thus, M outputs in every step a hypothesis j which is consistent with all the shortest strings seen so far. Now, another property of p_j comes into the play. Namely, M can effectively reproduce all information it has used to compute p_j from p_j. Therefore, when fed its last guess as well as the next input string, it can effectively decide whether or not it has to perform a mind change. This is the gist underlying the construction in Lange and Wiehagen (1991). We omit the details.

Furthermore, their result has some special features. They have shown that the IIM they actually use runs in time polynomial in the length of the input, since it totally avoids to test membership. The price paid is that it sometimes outputs inconsistent hypotheses. For a more detailed discussion concerning the consistent versus inconsistent learnability of the pattern languages the reader is referred to Wiehagen and Zeugmann (1994). ◇

Reviewing the results discussed so far the questions arises whether or not any indexed family learnable from text can be inferred by a conservative IIM or even by an iterative one. In each case, the answer is negative (cf. Lange and Zeugmann (1992),

(1993c)).

Theorem 4.

(1) $CCONSERVATIVE-TXT \subset ELIM-TXT$

(2) $CIT-TXT \subset ELIM-TXT$

Assertion (1) proves that overgeneralization is inevitable, in general, if one considers language learning from positive examples. In order to show a particular way to solve the subset problem, we discuss the relation between conservative learners and learning in the limit in some more detail. In Angluin's pioneering paper (cf. Angluin (1980b)) the following characterization of those indexed families for which learning in the limit from positive data is possible has been shown.

Proposition 3. (Angluin, 1980b) *Let $\mathcal{L}$ be an indexed family of recursive languages. Then: $\mathcal{L} \in ELIM-TXT$ if and only if there is an effective procedure which on every input $j \in \mathbb{N}$ enumerates a tell-tale set T_j of strings such that*

(1) *for all $j \in \mathbb{N}$, T_j is finite,*

(2) *for all $j \in \mathbb{N}$, $T_j \subseteq L_j$,*

(3) *for all $j, z \in \mathbb{N}$, if $T_j \subseteq L_z$, then $L_z \not\subset L_j$.*

From the characterization above one may deduce how the identification by enumeration principle has to be modified in order to obtain a suitable learning method for learning from positive data. In particular, this method provides insight into the problem of how to deal with overgeneralization.

Example 5. A universal IIM for learning from text

Let $\mathcal{L} \in ELIM-TXT$. Moreover, assume a corresponding procedure which on every input $j \in \mathbb{N}$ enumerates a tell-tale set T_j of strings satisfying the above requirements. Let $T_j^{(x)}$ denotes the finite subset of T_j which may be enumerated within x steps. The following IIM M $ELIM-TXT$–identifies $\mathcal{L}$. When fed an initial segment t_x of any text t for a language $L \in \mathcal{L}$, M searches for the least index $j \leq x$ satisfying $T_j^{(x)} \subseteq t_x^+ \subseteq L_j$. In case an index j is found, then M outputs the guess j. Otherwise, M requests the next input. $\diamond$

Again, the tell-tale sets are used to control the search within the hypothesis space in order to find a suitable consistent hypothesis. If the tell-tale set T_j is not completely enumerated within x steps, M may be forced to produce an overgeneralized hypothesis when fed a text t for L. Namely, after processing t_x it may happen that M outputs a hypothesis j with $L \subset L_j$. But, then there has to be a $y > x$ such that $T_j^{(y)}$ contains a string s not belonging to L (cf. Proposition 3, Condition (3)). Hence, s never appears in the text t for L and M will reject its former guess j. Furthermore, M will never output the guess j in any subsequent step.

Thus, we already know one way to design suitable enumeratively working IIMs that can solve every learning task from positive data as long as indexed families are concerned. However, a straightforward analysis shows that M is neither set-driven nor semantically finite. Hence, some desirable properties are still missing. We postpone the problem of order independence for a while and refer the reader to Section 8 for a detailed discussion. The remaining part of this section is devoted to the problem

whether or not semantical finiteness can always be achieved. Clearly, any conservative IIM is semantically finite, too. Therefore, let us have a closer look to conservative learning. We start with the following characterization of class comprising conservative inference.

Theorem 5. (Lange and Zeugmann, 1993d) *Let $\mathcal{L}$ be an indexed family. Then: $\mathcal{L} \in CCONSERVATIVE-TXT$ if and only if there are a hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ and a uniformly recursively generable family $(T_j)_{j\in\mathbb{N}}$ of finite non-empty sets such that*

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

(2) *for all $j \in \mathbb{N}$, $T_j \subseteq L(G_j)$,*

(3) *for all $j,\ z \in N$, if $T_j \subseteq L(G_z)$, then $L(G_z) \not\subset L(G_j)$.*

A family of finite sets $(T_j)_{j\in\mathbb{N}}$ is said to be ***uniformly recursively generable*** iff there is a total effective procedure g which, on every input j, generates all elements of T_j and stops.

Hence, the main difference between conservative learning and learning in the limit is characterized by the different degree of recursiveness of the corresponding tell-tale families. In case of conservative learning they are recursively generable, and in the general case of learning in the limit they are recursively enumerable. And indeed, recursive generability cannot always be achieved as our next example shows.

Example 6. The weakness of conservative learners

In order to verify $ELIM-TXT \setminus CCONSERVATIVE-TXT \neq \emptyset$ we define the following indexed family $\mathcal{L} = (L_{\langle j,k\rangle})_{k,j\in\mathbb{N}}$. For all $k \in \mathbb{N}$, let $L_{\langle k,0\rangle} = \{a^k b^n \mid n \in \mathbb{N}^+\}$. For all $k \in \mathbb{N}$ and all $j \in \mathbb{N}^+$,we distinguish the following cases:

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

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

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

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

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

$\mathcal{L} = (L_{\langle k,j\rangle})_{j,k\in\mathbb{N}}$ is an indexed family of recursive languages, since the predicate "$\Phi_i(y) \leq z$" is uniformly decidable in $i,\ y,$ and z. It is easy to see how a recursively enumerable tell-tale family has to be defined, and hence $\mathcal{L} \in ELIM-TXT$. Thus, it remains to ask whether or not $\mathcal{L}$ can be class comprisingly, and conservatively identified. An affirmative answer would imply that a finite tell-tale set for the infinite language $L_{\langle k,0\rangle}$ can be recursively generalized. But any procedure which recursively generates a corresponding tell-tale set may be used to solve simultaneously the halting problem, a contradiction. $\diamond$

Note that the same family may be used to witness the weakness of iterative IIMs, too. However, the reasons for $\mathcal{L} \notin CCONSERVATIVE-TXT$ and $\mathcal{L} \notin CIT-TXT$ are *different*. Conservative learners cannot handle overgeneralization at all, but iterative learners sometimes *can* (cf. Section 6, Theorem 18). The main reason for

$\mathcal{L} \notin CIT-TXT$ is the topological structure of the finite languages in $\mathcal{L}$. Every IIM learning $\mathcal{L}$ has to output an overgeneralized hypothesis. Taking this into account, it is intuitively clear that every iterative IIM fails to memorize the maximal element (with respect to lexicographical order) of some finite language a text of which it is fed. A comprehensive discussion concerning the power and limitations of iterative IIMs is provided in Section 6.

Up to now, we have discussed learning from text and learning from informant independently. Finally, let us mention an interesting aspect related to the interplay between information presentation and learnability constraints. In Lange and Zeugmann (1993a) it was shown that the family *PAT* is finitely identifiable from positive and negative examples. On the other hand, *PAT* is even learnable from text. Surprisingly enough, this observation can be generalized as follows.

Theorem 6. $SMON-INF \subseteq CONSERVATIVE-TXT$

Finally, we present a proof of the theorem above which is conceptually completely different from the one published in Lange and Zeugmann (1993a). It is mainly based on characterizations recently obtained (cf. Lange and Zeugmann (1992), (1994)).

Proof. The main ingredient is the following characterization of strong-monotonic inference from positive and negative data (cf. Lange and Zeugmann (1994)).

Theorem 7. *Let $\mathcal{L}$ be an indexed family of recursive languages. Then: $\mathcal{L} \in SMON-INF$ if and only if there are a hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ and uniformly recursively generable families $(P_j)_{j\in\mathbb{N}}$ and $(N_j)_{j\in\mathbb{N}}$ of finite sets such that*

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

(2) *for all $j \in \mathbb{N}$, $\emptyset \neq P_j \subseteq L(G_j)$ and $N_j \subseteq co-L(G_j)$,*

(3) *for all $j,\ z \in \mathbb{N}$, if $P_j \subseteq L(G_z)$ as well as $N_j \subseteq co-L(G_z)$, then $L(G_j) \subseteq L(G_z)$.*

Let $\mathcal{G}$ be the hypothesis space from the characterization above. We show that the family $(P_j)_{j\in\mathbb{N}}$ serves as a family of finite tell-tale sets satisfying the requirements in Theorem 5. In doing so, it suffices to show that $P_j \subseteq L(G_z)$ implies $L(G_z) \not\subset L(G_j)$ for all $j, z \in \mathbb{N}$. Suppose the converse, i.e., $P_j \subseteq L(G_z)$ as well as $L(G_z) \subset L(G_j)$ for any $j, z \in \mathbb{N}$.

Case 1. $N_j \cap L(G_z) = \emptyset$.

Thus, $P_j \subseteq L(G_z)$ as well as $N_j \subseteq co-L(G_z)$ implies $L(G_j) \subseteq L(G_z)$ by Theorem 7, Property (3). This contradicts $L(G_z) \subset L(G_j)$.

Case 2. $N_j \cap L(G_z) \neq \emptyset$.

Now, there has to be a string $s \in L(G_z)$ such that $s \in N_j \subseteq co-L(G_j)$. Therefore, $s \in L(G_z) \setminus L(G_j)$. This contradicts $L(G_z) \subset L(G_j)$.

The tell-tale family $(P_j)_{j\in\mathbb{N}}$ satisfies Conditions (2) and (3) in Theorem 5, if we select the class preserving hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$. From the proof of Theorem 5 (cf. Lange and Zeugmann (1993c)) it follows that $\mathcal{L}$ can be conservatively inferred with respect to the hypothesis space $\mathcal{G}$. Hence, $\mathcal{L} \in CONSERVATIVE-TXT$. q.e.d.

As we have seen, the learnability of indexed families provides several interesting problems that are worth to be studied in some more detail. The following sections survey recently obtained results in a systematic way.

4. Characterizations

As we have already seen, characterizations provide a useful tool to answer the question how learning algorithms may be designed (cf. Example 5). Moreover, they may help to gain a better understanding of what different learning types have in *common* and where the *differences* are (cf. Proposition 3 and Theorem 5). Furthermore, characterizations may be applied to solve particular learning problems and to solve deeper theoretical questions (cf. Example 4 and Theorem 6). Therefore, it is justified to deal with characterizations in some more detail. We continue our guided tour with a survey of representative results and proof techniques. For that purpose it suffices to deal with class preserving learning. We start with learning from positive and negative data.

4.1. Learning from Informant

Remembering our discussion in Section 3, we already know that Gold's (1967) *identification by enumeration principle* serves as a universal inference algorithm for learning in the limit. Moreover, every IIM realizing the identification by enumeration principle fulfills the weak-monotonic and dual weak-monotonic constraint. Hence, there is no need to characterize these learning models. On the other hand, neither $MON-INF$, MON^d-INF, $SMON-INF$ nor $SMON^d-INF$ is as powerful as learning in the limit. Hence, learning under these monotonicity constraints cannot be realized by a straightforward implementation of Gold's identification by enumeration principle. Therefore, we are interested in answering the question whether or not there is a universal inference method for these learning types. The affirmative answer is given by our next theorem that characterizes $MON-INF$ in terms of recursively generable finite tell-tale sets. Note that the same proof technique applies *mutatis mutandis* to all remaining learning types (cf. Lange and Zeugmann (1994)).

Theorem 8. *Let $\mathcal{L}$ be an indexed family of recursive languages. Then: $\mathcal{L} \in MON-INF$ if and only if there are a hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ and recursively generable families $(P_j)_{j\in\mathbb{N}}$ and $(N_j)_{j\in\mathbb{N}}$ of finite sets such that*

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

(2) *for all $j \in \mathbb{N}$, $\emptyset \neq P_j \subseteq L(G_j)$ and $N_j \subseteq co-L(G_j)$,*

(3) *for all $k, j \in \mathbb{N}$, and for all $L \in \mathcal{L}$, if $P_k \cup P_j \subseteq L(G_j) \cap L$ as well as $N_k \cup N_j \subseteq co-L(G_j) \cap co-L$, then $L(G_k) \cap L \subseteq L(G_j) \cap L$.*

Proof. Necessity: Let $\mathcal{L} \in MON$–INF. Then there are an IIM M and a hypothesis space $(\hat{G}_j)_{j\in\mathbb{N}}$ such that M infers every $L \in \mathcal{L}$ monotonically from any informant with respect to $(\hat{G}_j)_{j\in\mathbb{N}}$. Without loss of generality, we can assume that M is conservative, too, (cf. Lange and Zeugmann (1993a)). We proceed in showing how to construct $(\hat{G}_j)_{j\in\mathbb{N}}$. This is done in two steps. First, we define a hypothesis space $(\tilde{G}_j)_{j\in\mathbb{N}}$ as well as corresponding recursively generable families $(\tilde{P}_j)_{j\in\mathbb{N}}$ and $(\tilde{N}_j)_{j\in\mathbb{N}}$ of finite sets, where $\tilde{P}_j$ may be empty for some $j \in \mathbb{N}$. Afterwards, we define a procedure which enumerates a certain subset of $\mathcal{G}$.

First step: For all $k, x \in \mathbb{N}$ we set $\tilde{G}_{\langle k,x \rangle} = \hat{G}_k$. By construction, $range(\mathcal{L}) = \mathcal{L}(\tilde{\mathcal{G}})$ is obvious. Let i^k be the lexicographically ordered informant for $L(\hat{G}_k)$, and let $x \in \mathbb{N}$.

We define:

$$\tilde{P}_{\langle k,x \rangle} = \begin{cases} i^{k,+}_y, & \text{if} \quad y = min\{z \mid z \leq x,\ M(i^k_z) = k,\ i^{k,+}_z \neq \emptyset\}, \\ \emptyset, & \text{otherwise.} \end{cases}$$

If $\tilde{P}_{\langle k,x \rangle} = i^{k,+}_y \neq \emptyset$, then we set $\tilde{N}_{\langle k,x \rangle} = i^{k,-}_y$. Otherwise, we define $\tilde{N}_{\langle k,x \rangle} = \emptyset$.

The intuitive idea behind the definition of the families $(\tilde{P}_j)_{j \in \mathbb{N}}$ and $(\tilde{N}_j)_{j \in \mathbb{N}}$ is as follows. If an IIM M is fed initial segments of an informant then there is almost no hope to determine from the appearing outputs what the IIM M has learned from its input. However, in the particular case that we have an index k of the language L, the lexicographically ordered informant i^k of which is successively fed M, there is a syntactical criterion that can be effectively tested. Namely, if $M(i^k_z) = k$, then we know for sure that M has done a pretty good job on input i^k_z. Clearly, it has output a correct guess for the language it should learn. Therefore, it seems very reasonable to suitably collect all the information contained in these initial segments in the corresponding families $(\tilde{P}_j)_{j \in \mathbb{N}}$ and $(\tilde{N}_j)_{j \in \mathbb{N}}$. And indeed, it even suffices to collect the positive data in $\tilde{P}_j$, and the negative data in $\tilde{N}_j$.

Second step: The hypothesis space $(G_j)_{j \in \mathbb{N}}$ will be defined by simply striking off all grammars $\tilde{G}_{\langle k,x \rangle}$ with $\tilde{P}_{\langle k,x \rangle} = \emptyset$. In order to save readability, we omit the corresponding mapping yielding the enumeration $(G_j)_{j \in \mathbb{N}}$ from $(\tilde{G}_j)_{j \in \mathbb{N}}$. If G_j is referring to $\tilde{G}_{\langle k,x \rangle}$, we set $P_j = \tilde{P}_{\langle k,x \rangle}$ and $N_j = \tilde{N}_{\langle k,x \rangle}$.

We have to show that $(G_j)_{j \in \mathbb{N}}$, $(N_j)_{j \in \mathbb{N}}$, and $(P_j)_{j \in \mathbb{N}}$ do fulfill the announced properties. (1) and (2) follow immediately, since M has, in particular, to infer every $L \in \mathcal{L}$ from its lexicographically ordered informant. It remains to show (3). Suppose $L \in \mathcal{L}$ and $k,\ j \in \mathbb{N}$ such that $P_k \cup P_j \subseteq L(G_j) \cap L$ as well as $N_k \cup N_j \subseteq co - L(G_j) \cap co - L$. We have to show $L(G_k) \cap L \subseteq L(G_j) \cap L$. Due to our construction, we can make the following observations. There is a uniquely defined initial segment of the lexicographically ordered informant i^k for $L(G_k)$, say i^k_x, such that $range(i^k_x) = P_k \cup N_k$. Moreover, $M(i^k_x) = m$ with $L(G_k) = L(\hat{G}_m)$. By i^j_y we denote the uniquely defined initial segment of the lexicographically ordered informant i^j for $L(G_j)$ with $range(i^j_y) = P_j \cup N_j$. Furthermore, $M(i^j_y) = n$ and $L(G_j) = L(\hat{G}_n)$. From $P_k \subseteq L(G_j)$ and $N_k \subseteq co - L(G_j)$, it follows $i^k_x \sqsubseteq i^j$. Since $P_j \subseteq L$ and $N_j \subseteq co - L$, we conclude that i^j_y is an initial segment of the lexicographically ordered informant i^L for L.

We have to distinguish the following three cases.

Case 1. $x = y$

Hence, $m = n$ and therefore $L(G_k) = L(G_j)$. This implies $L(G_k) \cap L \subseteq L(G_j) \cap L$.

Case 2. $x < y$

Now, we have $i^k_x \sqsubseteq i^j_y \sqsubseteq i^L$. Moreover, M monotonically infers L from informant i^L. By the transitivity of "$\subseteq$" we immediately obtain $L(G_k) \cap L \subseteq L(G_j) \cap L$.

Case 3. $y < x$

Hence, $i^j_y \sqsubseteq i^k_x \sqsubseteq i^j$. Since M is conservative, too, it follows $m = n$. Therefore,

$L(G_k) = L(G_j)$. This implies $L(G_k) \cap L \subseteq L(G_j) \cap L$.

Hence, $(G_j)_{j \in \mathbb{N}}$, $(P_j)_{j \in \mathbb{N}}$ as well as $(N_j)_{j \in \mathbb{N}}$ have indeed the announced properties.

Sufficiency: It suffices to prove that there is an IIM M inferring any $L \in \mathcal{L}$ monotonically from any informant with respect to $\mathcal{G}$. Interestingly enough, an easy modification of the universal IIM described in Example 5 is all we need. So let $L \in \mathcal{L}$, let i be any informant for L, and $x \in \mathbb{N}$.

$M(t_x) =$ "Generate P_j and N_j for $j = 0, \ ..., \ x$ and test whether

$P_j \subseteq i_x^+ \subseteq L(G_j)$ and $N_j \subseteq i_x^- \subseteq co - L(G_j)$.

In case there is at least a j fulfilling the test, output the minimal one and request the next input. Otherwise, output nothing and request the next input."

Since all of the P_j and N_j are uniformly recursively generable and finite, we see that M is an IIM. We have to show that it infers L. Let z be the least k such that $L = L(G_k)$. We claim that M converges to z. Consider $P_0, \ ..., \ P_z$ as well as $N_0, \ ..., \ N_z$. Then there must be an x such that $P_z \subseteq i_x^+ \subseteq L(G_z)$ and $N_z \subseteq i_x^- \subseteq co - L(G_z)$. That means, at least after having fed i_x to M, the machine M outputs a hypothesis. Moreover, since $P_z \subseteq i_{x+r}^+ \subseteq L(G_z)$ as well as $N_z \subseteq i_{x+r}^- \subseteq co - L(G_z)$ for all $r \in \mathbb{N}$, the IIM M never produces a guess $j > z$ on i_{x+r}.

Suppose, M converges to $j < z$. Then we have: $P_j \subseteq i_{x+r}^+ \subseteq L(G_j) \neq L(G_z)$ and $N_j \subseteq i_{x+r}^- \subseteq co - L(G_j)$ for all $r \in \mathbb{N}$.

Case 1. $L(G_z) \setminus L(G_j) \neq \emptyset$

Consequently, there is at least one string $s \in L(G_z) \setminus L(G_j)$ such that$(s, +)$ has to appear sometime in i, say in i_{x+r} for some r. Thus, we have $i_{x+r}^+ \not\subseteq L(G_j)$, a contradiction.

Case 2. $L(G_j) \setminus L(G_z) \neq \emptyset$

Then we may restrict ourselves to the case $L(G_z) \subset L(G_j)$, since otherwise we are again in Case 1. Consequently, there is at least one string $s \in L(G_j) \setminus L(G_z)$ such that $(s, -)$ has to appear sometime in i, say in i_{x+r} for some r. Thus, $i_{x+r}^- \not\subseteq co - L(G_j)$, a contradiction.

Consequently, M converges to z on informant i. To complete the proof we show that M monotonically learns L. Suppose M outputs k and changes its mind to j in some subsequent step. Consequently, $M(i_x) = k$ and $M(i_{x+r}) = j$, for some $x, \ r \in \mathbb{N}$.

Case 1. $L(G_j) = L$

Hence, $L(G_k) \cap L \subseteq L(G_j) \cap L = L$ is obviously fulfilled.

Case 2. $L(G_j) \neq L$

Due to the definition of M, it holds $P_k \subseteq i_x^+ \subseteq i_{x+r}^+ \subseteq L(G_j)$. Hence, $P_k \subseteq L \cap L(G_j)$. Furthermore, we have $N_k \subseteq i_x^- \subseteq i_{x+r}^- \subseteq co - L(G_j)$. This implies $N_k \subseteq co - L(G_j) \cap co - L$. Since $M(i_{x+r}) = j$, it holds that $P_j \subseteq L$ and $N_j \subseteq co - L$. This yields $P_k \cup P_j \subseteq L(G_j) \cap L$ as well as $N_k \cup N_j \subseteq co - L(G_j) \cap co - L$. From (3), we obtain $L(G_k) \cap L \subseteq L(G_j) \cap L$.

Hence, M $MON{-}INF$–identifies $\mathcal{L}$. q.e.d.

As a matter of fact, the machine defined above uses the tell-tale sets to control its search within the hypothesis space. Therefore, the desired modification of the identification by enumeration principles has been obtained using the same ideas that we have exemplified in the proof of Theorem 1 in Section 3.

Exploiting the same proof method similar characterizations for MON^d-INF, $SMON-INF$, and $SMON^d-INF$ have been obtained (cf. Lange and Zeugmann (1994)). Hence, the different monotonicity constraints are completely characterized by the specific properties of the relevant recursively generable tell-tale families. Note that the resulting characterization for $SMON-INF$ is stated in Section 3, Theorem 7. Finally, the characterization of LIM_k-INF required some new ingredients (cf. Lange and Zeugmann (1993a)). However, the difficulties one has to overcome when characterizing LIM_k-INF are closely related to the problems one has to handle in characterizing LIM_k-TXT. Therefore, we refer the reader to Theorem 10 below.

4.2. Learning from Text

When investigating monotonic language learning from text, the situation is much more subtle. As we shall see later, the IIM M defined in the proof of Theorem 8 does not longer serve as a universal learning device. Consequently, a relaxation of the basic approach underlying $M's$ definition is necessary. We shall come back to this point.

Although Angluin (1980b) established some sufficient conditions that guarantee exact conservative learning from positive data, it remained open whether the class of those indexed families for which exact conservative learning from positive data is possible may be characterized in terms of finite non-empty tell-tale sets. Next, we present a preliminary solution to this long standing open problem. In doing so, we characterize $WMON-TXT$ in terms of recursively generable finite tell-tales. The underlying proof method is powerful enough to successfully attack the original problem, too, (cf. Section 8, Theorem 42).

Theorem 9. *Let $\mathcal{L}$ be an indexed family. Then: $\mathcal{L} \in WMON-TXT$ if and only if there are a hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ and a recursively generable family $(T_j)_{j\in\mathbb{N}}$ of finite and non-empty sets such that*

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

(2) *for all $j \in \mathbb{N}$, $T_j \subseteq L(G_j)$,*

(3) *for all $j,\ z \in \mathbb{N}$, if $T_j \subseteq L(G_z)$, then $L(G_z) \not\subset L(G_j)$.*

Proof. Necessity: Let $\mathcal{L} \in WMON-TXT = CONSERVATIVE\text{–}TXT$. Then there are an IIM M and a hypothesis space $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ such that M conservatively infers every $L \in \mathcal{L}$ with respect to $\hat{\mathcal{G}}$. The desired hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ and the corresponding tell-tale family $(T_j)_{j\in\mathbb{N}}$ can be defined in a similar way as in the demonstration of Theorem 8. Therefore, we point to the differences, only. First, we construct a hypothesis space $\tilde{\mathcal{G}} = (\tilde{G}_j)_{j\in\mathbb{N}}$ as well as a recursively generable family $(\tilde{T}_j)_{j\in\mathbb{N}}$ of finite but possibly empty sets. For all $k,\ x \in \mathbb{N}$, we set $\tilde{G}_{\langle k,x\rangle} = \hat{G}_k$. Furthermore, for any language $L(\hat{G}_k)$, let t^k be the canonically ordered text of $L(\hat{G}_k)$.

We define:

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

Obviously, $(\tilde{T}_j)_{j\in\mathbb{N}}$ is a uniformly recursively generable family of finite sets.

Conceptually, we use the same idea as in the proof of Theorem 8. The main new ingredient is the introduction of the *canonical* text. Clearly, if $L(\hat{G}_k)$ is finite, then the sequence of its lexicographically ordered strings is finite, too. Thus, it does not constitute a text for $L(\hat{G}_k)$. The tempting idea to repeat the lexicographically largest string infinitely often fails, since the resulting text becomes non-recursive. As a consequence, the tell-tales were no longer uniformly recursively generable. Using the canonical text, all these difficulties vanish.

The desired hypothesis space $\mathcal{G}$ is obtained from $\tilde{\mathcal{G}}$ by simply striking off all grammars $\tilde{G}_{\langle k,x\rangle}$ for which $\tilde{T}_{\langle k,x\rangle} = \emptyset$. Analogously, $(T_j)_{j\in\mathbb{N}}$ is obtained from $(\tilde{T}_j)_{j\in\mathbb{N}}$. Obviously, $(T_j)_{j\in\mathbb{N}}$ is a recursively generable family of finite and non-empty sets. In order to save notational convenience, we refer to T_j as to $T_{\langle k,x\rangle}$, i.e., we omit the corresponding mapping yielding the enumeration of the sets T_j from $\tilde{T}_z$. It remains to show that $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ and $(T_j)_{j\in\mathbb{N}}$ do fulfill the announced properties. Due to our construction, (2) holds obviously. In order to prove (1), let $L \in \mathcal{L}$. We have to show that there is at least a $j \in \mathbb{N}$ such that for $j = \langle k, x\rangle$ we have $L = L(G_{\langle k,x\rangle})$. For this purpose, due to our construction, it suffices to show that $\tilde{T}_{\langle k,x\rangle} \neq \emptyset$. Let t be L's canonically ordered text. Since M has to infer L from t, there are $k,\ y \in \mathbb{N}$ such that for all $z < y,\ M(t_z) \neq k,\ M(t_y) = k$ and $L = L(\hat{G}_k)$. Consequently, $\tilde{T}_{\langle k,y\rangle} = t_y^+$. Hence, by the convention made above, we get that $T_{\langle k,y\rangle} = t_y^+$. Moreover, it immediately follows that $L = L(G_{\langle k,x\rangle})$ for any $x \geq y$. This proves Property (1).

Finally, we have to show (3). It results from the requirement that any conservative IIM is never allowed to output an overgeneralized hypothesis. To see this, suppose the converse, i.e., there are $j,\ z \in \mathbb{N}$ such that $T_j \subseteq L(G_z)$ and $L(G_z) \subset L(G_j)$. By definition, there are uniquely determined $k, x \in \mathbb{N}$ such that $j = \langle k, x\rangle$. Let $s_0, ..., s_y$ be the sequence of strings of T_j in canonical order with respect to $L(G_{\langle k,x\rangle})$ such that $M(s_0, ..., s_y) = k$. Now we conclude that $s_0, ..., s_y$ is an initial segment of the canonically ordered text for $L(G_z)$, since $T_j \subseteq L(G_z) \subset L(G_j) = L(G_{\langle k,x\rangle})$. Finally, M has to infer $L(G_z)$ from its canonically ordered text. Thus, it has to perform a mind change in some subsequent step which cannot be caused by an inconsistency. This contradiction yields (3).

Sufficiency: It suffices to prove that there is a conservative IIM M inferring every $L \in \mathcal{L}$ from text with respect to $\mathcal{G}$. First, we slightly modify the corresponding tell-tale family. For all $j \in \mathbb{N}$, we set $\hat{T}_j = \bigcup_{n\leq j} T_n \cap L(G_j)$. Note that the new tell-tale family fulfills Properties (1) through (3). Let $L \in \mathcal{L}$, $t \in text(L)$, and $x \in \mathbb{N}$.

$M(t_x) =$ "For $j = 0,\ ...,\ x$, generate $\hat{T}_j$ and test whether $\hat{T}_j \subseteq t_x^+ \subseteq L(G_j)$.

In case there is one j fulfilling the test, output the minimal one and request the next input. Otherwise, output nothing and request the next input."

Since all of the $\hat{T}_j$ are uniformly recursively generable and finite, we see that M is an IIM. Now it suffices to show that M conservatively infers L from t.

Claim 1. M is conservative.

Let k and j be two hypotheses produced by M on input t_x and t_{x+r}, respectively. We have to show that $t^+_{x+r} \not\subseteq L(G_k)$. For that purpose we distinguish the following cases.

Case 1. $k < j$

Due to M's definition we immediately obtain $t^+_{x+r} \not\subseteq L(G_k)$.

Case 2. $j < k$

Suppose, $t^+_{x+r} \subseteq L(G_k)$. In accordance with its definition, M has verified that $\hat{T}_j \subseteq t^+_{x+r} \subseteq L(G_j)$. Moreover, the definition of the tell-tale family directly yields $\hat{T}_j \subseteq \hat{T}_k$, since $j < k$ and $\hat{T}_j \subseteq t^+_{x+r} \subseteq L(G_k)$. Taking into account that $\hat{T}_k \subseteq t^+_x$, this implies $\hat{T}_j \subseteq t^+_x \subseteq L(G_j)$. Finally, since $j < k$ we conclude $M(t_x) = j$, a contradiction. Hence, the claim is proved.

Claim 2. M infers L from t.

Let z be the least k such that $L(G_k) = L$. Therefore, $L(G_j) \neq L$ for all $j \leq z$. By Property (3) we obtain that $L \backslash L(G_j) \neq \emptyset$ for all $j < z$ provided $\hat{T}_j \subseteq L$. Consequently, every candidate hypothesis $j < z$ is sometimes rejected by M, and M converges to z. Hence, the claim follows.

This proves the theorem. q.e.d.

Looking at the definition of the IIM M it is easy to see that this machine is conceptually the same as that one used in the demonstration of Theorem 8. Since M is conservative, M it is semantically finite, too. Moreover, M's output exclusively depends on the length as well as on the range of its input. IIMs satisfying the latter requirement are called *rearrangement-independent* (cf. Definition 12, Section 8).

The proof method explained above applies *mutatis mutandis* to characterize strong-monotonic as well as dual strong-monotonic inference from positive data. All these characterizations express the different monotonicity constraints the relevant learner has to fulfill by a specific modification of Property (3) from the latter theorem (cf. Lange and Zeugmann (1992)). Hence, we again arrived at a *unifying framework*. As a consequence, class preserving strong-monotonic as well as dual strong-monotonic inference can always be realized by a semantically finite and rearrangement-independent IIM.

Looking at monotonic language learning from text, the situation considerably changes (cf. Section 8, Theorem 41). Thus, the IIM M defined in the demonstration of Theorem 9 does not serve as universal learning algorithm for monotonic and dual monotonic inference from text. The same difficulties arise when dealing with learning within an *a priori* fixed number of mind changes. Therefore, we continue with a characterization of $LIM_k{-}TXT$, and explain the new proof technique. Note that the following theorem establishes a new method how to handle overgeneralization. This solution to the subset problem considerably improves Angluin's method (cf. Example 5). Finally, the main idea used in the following proof is powerful enough to establish characterizations for monotonic and dual monotonic learning, respectively (cf. Zeugmann, Lange and Kapur (1995)).

In order to characterize $LIM_k{-}TXT$ in terms of recursively generable finite tell-

tale sets we have been forced to define an easily computable relation $\prec \subseteq \mathbb{N} \times \mathbb{N}$ that can be used to distinguish appropriate chains of tell-tales with the help of which an IIM M may compute its hypotheses. Now we are ready to present the desired characterization.

Theorem 10. *Let $\mathcal{L}$ be an indexed family, and $k \in \mathbb{N}^+$. Then: $\mathcal{L} \in LIM_k{-}TXT$ if and only if there are a hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$, a computable relation $\prec$ over $\mathbb{N}$, and a recursively generable family $(T_j)_{j\in\mathbb{N}}$ of finite and non-empty sets such that*

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

(2) *for all $L \in \mathcal{L}$ and all $z \in \mathbb{N}$, $T_z \subseteq L(G_z)$,*

(3) *for all $L \in \mathcal{L}$ and every $z \in \mathbb{N}$, and all finite $A \subseteq L$, if $T_z \subseteq L$, $L(G_z) \neq L$, then there is a j such that $z \prec j$, and $A \subseteq T_j \subseteq L(G_j) = L$,*

(4) *for all $L \in \mathcal{L}$, there is no sequence $(z_j)_{j=0,\ldots,k+1}$ such that for all $j \leq k$, $z_j \prec z_{j+1}$ as well as all $T_{z_j} \subseteq T_{z_{j+1}} \subseteq L$*

Proof. Necessity: Without loss of generality, let M be an IIM consistently inferring $\mathcal{L}$ with respect to some hypothesis space $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ (cf. Lange and Zeugmann (1993b)). Then, for every $L \in \mathcal{L}$ and every $t \in text(L)$ the IIM M performs at most k mind changes when successively fed t. First, we construct a hypothesis space $\tilde{\mathcal{G}} = (\tilde{G}_j)_{j\in\mathbb{N}}$ as well as a recursively generable family $(\tilde{T}_j)_{j\in\mathbb{N}}$ of finite but possibly empty sets. Then, we describe a procedure enumerating a certain subset of $\tilde{\mathcal{G}}$ which we call $\mathcal{G}$. Finally, we define the desired relation $\prec$.

Let $\sigma_0, \sigma_1, \sigma_2, \ldots$ be an effective enumeration of all finite, non-null sequences of strings from Σ^* such that $\sigma_x \sqsubset \sigma_y$ implies $x < y$ for all $x, y \in \mathbb{N}$. Furthermore, for all $n, x \in \mathbb{N}$ we set $\tilde{G}_{\langle n,x\rangle} = \hat{G}_n$. The family $(\tilde{T}_{\langle n,x\rangle})_{n,x\in\mathbb{N}}$ is defined as follows.

$$\tilde{T}_{\langle n,x\rangle} = \begin{cases} \sigma_x^+, & \text{if } M(\sigma_x) = n, \\ \emptyset, & \text{otherwise.} \end{cases}$$

Obviously, $(\tilde{T}_{\langle n,x\rangle})_{n,x\in\mathbb{N}}$ is a uniformly recursively generable family of finite sets. Furthermore, by construction we have $\mathcal{L}(\tilde{\mathcal{G}}) = range(\mathcal{L})$.

Claim 1. For all $L \in \mathcal{L}$ there exists an index $\langle n, x\rangle$ such that $\tilde{T}_{\langle n,x\rangle} \neq \emptyset$ and $L(\tilde{G}_{\langle n,x\rangle}) = L$.

Let t be the canonical text of L. Since M learns L, there exist $n, y \in \mathbb{N}$ such that $M(t_y) = n$ and $L = L(\hat{G}_n)$. Moreover, t_y is a finite, non-null sequence. Hence, there has to be an x such that $\sigma_x = t_y$. Consequently, $\tilde{T}_{\langle n,x\rangle} \neq \emptyset$, and $L(\tilde{G}_{\langle n,x\rangle}) = L$. This proves the claim.

We proceed with the definition of the desired hypothesis space $\mathcal{G}$ and the relation $\prec$. For that purpose we define a recursive function f as follows. Let $f(0)$ be the least j with $\tilde{T}_j \neq \emptyset$, and for all $j \geq 1$ let

$$f(j) = \begin{cases} j, & \text{if } \tilde{T}_j \neq \emptyset, \\ f(j-1), & \text{otherwise.} \end{cases}$$

Furthermore, we define $G_j = \tilde{G}_{f(j)}$ and $T_j = \tilde{T}_{f(j)}$ for all $j \in \mathbb{N}$. Finally, let $z, j \in \mathbb{N}$, and let m, n, x, y be the uniquely determined numbers such that $f(z) = \langle m, y \rangle$ and $f(j) = \langle n, x \rangle$. Then we define $z \prec j$ if and only if $m \neq n$ and $\sigma_y \sqsubset \sigma_x$.

Clearly, $(T_j)_{j \in \mathbb{N}}$ is a uniformly recursively generable family of finite and non-empty sets and the relation $\prec$ is computable. It remains to show that Properties(1) through (4) are satisfied. Property (1) is an immediate consequence of Claim 1 and the definition of $\mathcal{G}$. In order to prove (2) it suffices to show that $n = M(\sigma_x)$ implies $\sigma_x^+ \subseteq L(\hat{G}_n)$. But this is obvious, since M is a consistent IIM.

We continue in proving Property (3). Let $L \in \mathcal{L}$, let $A \subseteq L$ be any finite set, and let $z \in \mathbb{N}$ be any index such that $T_z \subseteq L$ and $L(G_z) \neq L$. We have to show that there is a j with $z \prec j$ and $A \subseteq T_j \subseteq L(G_j) = L$. In accordance with our construction we have $T_z = \tilde{T}_{f(z)}$ and $G_z = \tilde{G}_{f(z)}$. Let m, y be the uniquely determined numbers with $f(z) = \langle m, y \rangle$. Then we know that $M(\sigma_y) = m$ and $L \neq L(\hat{G}_m)$, since $L(\hat{G}_m) = L(\tilde{G}_{\langle m,y \rangle}) = L(G_z) \neq L$. Moreover, $T_z = \sigma_y^+ \subseteq L$. Hence, σ_y is an initial segment of a text for L. Since $L \neq L(G_z)$, we additionally know that M, on input σ_y, has not yet converged to a correct hypothesis for L. Now, let t be the canonical text of L. Since $A \subseteq L$, there exists an $a \in \mathbb{N}$ such that $A \subseteq t_a^+$. Moreover, M has to learn L from every text for it. Consequently, there has to be an $r \in \mathbb{N}$ such that for $n = M(\sigma_y t_{a+r})$ the condition $L(\hat{G}_n) = L$ is satisfied. Furthermore, since $\sigma_y t_{a+r}$ is a finite sequence, there exits an index x with $\sigma_x = \sigma_y t_{a+r}$. By construction we get $\emptyset \neq \tilde{T}_{\langle n,x \rangle} \subseteq L(\tilde{G}_{\langle n,x \rangle}) = L$. Thus, there is a number j such that $f(j) = \langle n, x \rangle$, and for every j with $f(j) = \langle n, x \rangle$ we get $\sigma_y \sqsubset \sigma_x$ and $m \neq n$. Therefore, $z \prec j$, and (3) is proved.

We proceed with the demonstration of (4). Looking at the definition of the relation $\prec$ we immediately realize that $z \prec j$ implies $T_z \subseteq T_j$. Suppose there is a finite sequence $(z_j)_{j=0,\ldots,k+1}$ such that for all $j \leq k$, $z_j \prec z_{j+1}$ and $T_{z_j} \subseteq T_{z_{j+1}} \subseteq L$. Since $z_j \prec z_{j+1}$ and $T_{z_j} \subseteq T_{z_{j+1}} \subseteq L$, we get an initial segment of a text t of L on which M changes its mind $k+1$ times, a contradiction. Hence, (4) is proved.

Sufficiency: Again, it suffices to describe an IIM M that infers $\mathcal{L}$ in the limit with at most k mind changes from text with respect to $\mathcal{G}$. Let $L \in \mathcal{L}$, let t be any text for L, and let $x \in \mathbb{N}$. We define the desired IIM M as follows:

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

(A) Search for the least $j \leq x$ such that $T_j \subseteq t_x^+$. In case it is found, set $y_j = x$, output j and request the next input. Otherwise, output nothing and request the next input.

(B) Let $j = M(t_{x-1})$. Test whether there exists a $z \leq x$ such that $j \prec z$ and $t_{y_j}^+ \subseteq T_z \subseteq t_x^+$. In case one z is found, set $y_z = x$, output z and request the next input. Otherwise, output j and request the next input."

Since all of the T_j are uniformly recursively generable and finite and since $\prec$ is computable, we directly obtain that M is an IIM. We proceed in showing that M identifies L from t with at most k mind changes.

Claim 1. M converges and $card(\{x \mid M(t_x) \neq M(t_{x+1}\}) \leq k$.

Because of (1) and (2), M generates at least one hypothesis when fed the text t. Furthermore, assume for a moment that M performs more than k mind changes when inferring L from t. It is easy to recognize that this assumption would imply the existence of a sequence $(T_{z_j})_{j=0,\ldots,m}$ with $m > k$ such that for all $j < m$, $z_j \prec z_{j+1}$, and $T_{z_j} \subseteq T_{z_{j+1}} \subseteq L$. This would contradict (4). Thus, the number of possible mind changes M may perform when fed t is bounded by k. Moreover, M outputs after a certain period always a hypothesis. Hence, we may conclude that M converges.

Claim 2. If M converges, then the hypothesis M converges to is correct.

Assume that M converges to a hypothesis z with $L(G_z) \neq L$. Let x be the least index such that $M(t_x) = z$. Note that $T_z \subseteq t_x^+$ by M's definition. By Property (3), there has to be a j such that $z \prec j$, $t_x^+ \subseteq L(G_j)$, and $L(G_j) = L$. Hence, M performs an additional mind change when fed a sufficiently large initial segment t_x of t satisfying $T_j \subseteq t_x^+$, a contradiction.

By Claim 1 and 2, M infers any $L \in \mathcal{L}$ with at most k mind changes. Thus, the theorem is proved. q.e.d.

Note that the IIM defined in the proof of Theorem 10 uses a new technique to detect whether or not it has to perform a mind change. Clearly, no IIM can prove that its actual guess is correct, except in case it finitely learns. Hence, the machine has to collect evidence allowing it to decide whether it should prefer a new guess instead of maintaining its actual one. The machine defined in the proof above achieves this goal by using *a priori* knowledge concerning the hypothesis space as well as concerning the family of tell-tale sets. This *a priori* knowledge is provided by the computable relation. We believe that this approach considerably refines Angluin's (1980b) method how to detect overgeneralization. Finally, it should be mentioned that a conceptually similar, but technically different approach has been successfully applied to limit learning of recursive functions (cf. Wiehagen (1991)).

5. Learnability in Dependence on the Hypothesis Space

Historically, most authors have investigated exact learnability. Moreover, many investigations in other domains of algorithmic learning theory deal with exact learning too (cf. e.g. Natarajan (1991)). And indeed, as long as one considers learning in the limit without any additional demand, every indexed family that is class comprisingly learnable may be exactly inferred, too (cf. Lange and Zeugmann (1993b)). However, when dealing with characterizations it turned out to be very helpful to construct class preserving hypothesis spaces (cf. Kapur and Bilardi (1992), Lange and Zeugmann (1992)). Consequently, it is only natural to ask whether or not class preserving learning algorithms are more powerful than exact ones. Dealing with a measure of efficiency we found that an appropriate choice of a class preserving hypothesis space may eventually increase the learning power (cf. Lange and Zeugmann (1993b)). Recently, these results have been considerably improved and similar effects concerning the relation between class preserving and class comprising inference have been elaborated (cf. Lange (1994)). Furthermore, studying the capabilities of learning algorithms in dependence on the hypothesis space has yield very interesting results concerning

probabilistic learning models (cf., e.g., Anthony and Biggs (1992), Freivalds, Kinber and Wiehagen (1988)). Therefore, it is worth to study this phenomenon in some more detail.

First, we present results demonstrating the superiority of class comprising to class preserving monotonic learning algorithms that are themselves superior to exact ones. These separations have been obtained by developing a *new powerful proof technique*. Establishing the announced separations using standard proof techniques would require to diagonalize against all hypothesis spaces and all IIMs. Instead, we have elaborated an *almost always effective reduction* of the halting problem to monotonic learning problems. This approach yields easy to describe indexed families witnessing the desired separations. Next, we present results comparing class comprising and exact inference procedures. These results strongly recommend the designer of learning algorithms to carefully choose the enumeration as well as the description of the target languages to get exact learning procedures of maximal power (cf. Theorem 15).

Finally, we ask why, for example, class preserving inference is sometimes more powerful than exact learning. Obviously, as long as there is an effective compiler from the space $\mathcal{G}$ of hypotheses into the indexed family $\mathcal{L}$, both models of inference are of equal power. Looking at learning in the limit, Gold (1967) proved that even limiting recursive compilers do suffice. What we present is a characterization result stating that exact learning is of the same power as class preserving inference if and only if there is a limiting recursive compiler satisfying an appropriate monotonicity requirement (cf. Theorem 17). Hence, our separations prove the non-existence of such compilers.

5.1. Separations

In this subsection we compare the learning capabilities of most of the introduced models of monotonic and dual-monotonic inference in dependence on the hypothesis space. The underlying selection aims to illustrate different effects which occur. The following theorem provides a summary of results obtained that relate the power of exact identification, class preserving inference, and class comprising learning under various monotonicity constraints to one another (cf. Lange, Zeugmann and Kapur (1992), Lange and Zeugmann (1993c, 1993d)).

Theorem 11.

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

For instance, it turns out that dual weak-monotonic learning is exactly as powerful as learning in the limit, if class comprising hypothesis spaces are admissible. In particular, a dual weak-monotonic learner may realize a suitable interplay between generalization and specialization (cf. Theorem 15). In comparison to $CWMON-TXT \subset CLIM-TXT$, having the freedom to combine both, generalization and specialization is essential in order to achieve maximal learning power. We consider this as a particular answer to the long-standing debate in the machine learning community for and against learning by generalization and learning by specialization, respectively.

Furthermore, the following incomparabilities have been shown (cf. Lange and Zeugmann (1993c, 1993d)).

Theorem 12.

(1) $WMON-TXT \ \# \ CSMON-TXT$

(2) $EWMON-TXT \ \# \ CSMON-TXT$

(3) $EWMON-TXT \ \# \ SMON-TXT$

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

Detailed proofs of both theorems can be found in Lange and Zeugmann (1993d). Because of the lack of space, we demonstrate one separation, only. On the one hand, this should illustrate the underlying proof technique. On the other hand, the corresponding result has some special features distinguishing it from most of the other ones (cf. Theorem 14).

Theorem 13. $SMON^d-TXT \subset CSMON^d-TXT$

Proof. Obviously, by definition $SMON^d-TXT \subseteq CSMON^d-TXT$. It suffices to show that $CSMON^d-TXT \setminus SMON^d-TXT \neq \emptyset$. The desired indexed family $\mathcal{L}_{csd} = (L_{\langle k,j\rangle})_{k,j\in\mathbb{N}}$ is defined as follows. For all $k \in \mathbb{N}$ we set $L_{\langle k,0\rangle} = \{a^{k+1}\} \cup \{b^n \mid n \in \mathbb{N}^+\}$. For all $k \in \mathbb{N}$ and all $j \geq 1$, we distinguish the following cases:

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

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

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

Then, we set $L_{\langle k,j\rangle} = \{a^{k+1}\} \cup \{b^m \mid 1 \leq m \leq x\} \cup \{c^j\}$.

Lemma 1. $\mathcal{L}_{csd} \notin SMON^d-TXT$

Since $EFIN-TXT = SMON^d-TXT$ (cf. Theorem 11), it remains to show that there is no IIM M which finitely infers $\mathcal{L}_{csd}$ with respect to $\mathcal{L}_{csd}$. Thereby, we effectively reduce the halting problem to $\mathcal{L}_{csd} \in EFIN-TXT$.

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

Let M be any IIM witnessing $\mathcal{L}_{csd} \in EFIN-TXT$. First of all, we define an algorithm $\mathcal{A}$ solving the halting problem. On input $k \in \mathbb{N}$ the algorithm $\mathcal{A}$ executes the following instructions:

(A1) For $z = 0, 1, 2, \ldots$ generate successively the lexicographically ordered text t of $L_{\langle k,0\rangle}$ until M on input t_z outputs for the first time a hypothesis of the form $\langle k, j\rangle$, i.e., $M(t_z) = \langle k, j\rangle$.

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

Due to our assumption, M in particular finitely infers $L_{\langle k,0\rangle}$ from its lexicographically ordered text. Taking $\mathcal{L}_{csd}$'s definition into account one can easily deduce that M has to output a hypothesis of the form $\langle k, j\rangle$, since any other hypotheses describe a language which is definitely different from $L_{\langle k,0\rangle}$. Thus, Instruction (A1) has to terminate. Due to the definition of a complexity measure Instruction (A2) can be effectively accomplished. Therefore, algorithm $\mathcal{A}$ eventually terminates for every number k.

It remains to show that $\varphi_k(k)$ is undefined, if $\neg\, \Phi_k(k) \leq max\{j, z\}$. Suppose the converse, i.e., $\varphi_k(k)$ is defined. Then, $\Phi_k(k) = x$ for some $x > max\{z, j\}$. Taking again $\mathcal{L}_{csd}$ definition into account it follows $L_{\langle k,j\rangle} = L_{\langle k,0\rangle}$, since $x > j$. Now, let m be the maximal index such that $b^m \in t_z^+$. Obviously, $m < x$, since t is the lexicographically ordered text of $L_{\langle k,0\rangle}$. Thus, t_z is an initial segment of a text for the finite language $L_{\langle k,x\rangle}$. Note that $L_{\langle k,x\rangle} \neq L_{\langle k,0\rangle}$. Thus, M fails to finitely infer $L_{\langle k,x\rangle}$ from every text having the initial segment t_z. This contradiction completes the proof of the above claim. Hence, Lemma 1 follows.

Lemma 2. $\mathcal{L}_{csd} \in CSMON^d\text{–}TXT$

We have to show that there are an appropriate hypothesis space $\mathcal{G}_{csd}$ comprising $\mathcal{L}_{csd}$ and an IIM M inferring $\mathcal{L}$ dual strong-monotonically with respect to $\mathcal{G}_{csd}$.

We define the wanted hypothesis space $\mathcal{G}_{csd} = (G_{\langle k,j\rangle})_{k,j\in\mathbb{N}}$ as follows. For all $k,\ j \in \mathbb{N}$, we set $L(G_{\langle k,0\rangle}) = \bigcup\limits_{j\in\mathbb{N}} L_{\langle k,j\rangle}$ and $L(G_{\langle k,j+1\rangle}) = L_{\langle k,j\rangle}$. Taking the definition of $\mathcal{L}_{csd}$ into account, it is easy to verify that membership is uniformly decidable for $\mathcal{G}_{csd}$.

Now, let $k \in \mathbb{N}$. If $\varphi_k(k)$ is undefined, we have $L(G_{\langle k,0\rangle}) = L_{\langle k,j\rangle} = L_{\langle k,0\rangle}$ for all $j \in \mathbb{N}$. Otherwise, i.e., if $\varphi_k(k)$ is defined, we have $L(G_{\langle k,0\rangle}) \supset L_{\langle k,j\rangle}$ for all $j \in \mathbb{N}$. Thus, a dual strong-monotonic learner M may simply output the hypothesis $\langle k, 0\rangle$ after the corresponding identifier a^{k+1} has been presented. No matter whether or not $\varphi_k(k)$ is defined, $\langle k, 0\rangle$ is a suitable hypothesis. If $\varphi_k(k)$ is undefined, $\langle k, 0\rangle$ is already the desired final guess. Otherwise, one string has to appear which tells M in which way the language $L(G_{\langle k,0\rangle})$ has to be specialized. Consequently, M $CSMON^d\text{–}TXT$–identifies $\mathcal{L}_{csd}$ with respect to the class comprising hypothesis space $\mathcal{G}_{csd}$. q.e.d.

The *new proof technique* demonstrated above applies to obtain most of the stated separations. In particular, we can almost always effectively reduce the halting problem to several monotonic learning problems. These reductions imply that the considered learning problems are at least as hard as the halting problem. This insight deserves some attention. Gold (1967) showed that no IIM can learn the class $\mathcal{R}$ of all recursive functions in the limit. On the other hand, the degree of the algorithmic unsolvability of $\mathcal{R} \in LIM$ is strictly less than the degree of the halting problem (cf., Adleman and Blum (1991)). This puts the constraint to learn monotonically with respect to a particular hypothesis space into a new perspective. An algorithmically solvable learning problem (e.g. $\mathcal{L}_{csd} \in CSMON^d\text{–}TXT$) may become algorithmically unsolvable, if an

at first glance natural demand is added (e.g. to learn class preservingly). Moreover, the degree of unsolvability may be at least as high as that of the halting problem, and is, therefore, strictly higher than that of learning all recursive functions. As far as we know, there is only one paper stating an analogous result in the setting of inductive inference of recursive functions, namely Freivalds, Kinber and Wiehagen (1992).

5.2. Class Comprising and Exact Learning

When dealing with monotonic inference, class comprising learning is almost always more powerful than class preserving inference which itself is superior to exact learning. In particular, the results obtained give strong evidence that exclusively changing the descriptions for the objects to be learned as well as their enumeration does not suffice to get learning algorithms of maximal power. Therefore, we are interested in knowing what kind of languages has to be supplemented to hypothesis spaces in order to design superior inference procedures. Moreover, we ask whether or not these added languages may be learned themselves as well. As the following theorems show, the answer to these questions strongly depends on the type of monotonicity requirement involved.

First, we investigate dual-strong monotonic learning. Applying the same idea used in the demonstration of $\mathcal{L}_{csd} \notin EFIN{-}TXT$ (cf. Theorem 13, Lemma 1) one can easily show that $\mathcal{L}(\mathcal{G}_{csd}) \notin SMON^d{-}TXT$. Hence, in order to learn the family $\mathcal{L}_{csd}$ strong-monotonically one is required to add grammars to the hypothesis space that describe languages being *not learnable themselves*. It turns out that this property is characteristic for dual strong-monotonic inference as the following theorem shows.

Theorem 14. *Let $\mathcal{L}$ be any indexed family satisfying $\mathcal{L} \in CSMON{-}TXT^d \setminus SMON^d{-}TXT$. Then there is no hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ such that $\mathcal{L} \subset \mathcal{L}(\mathcal{G})$, and $\mathcal{L}(\mathcal{G}) \in SMON^d{-}TXT$.*

Proof. Suppose the converse, i.e., there is a hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ such that $\mathcal{L}(\mathcal{G}) \in SMON^d{-}TXT$. By assumption, $\mathcal{L} \in CSMON^d{-}TXT \setminus SMON^d{-}TXT$, and hence, $\mathcal{L} \subset \mathcal{L}(\mathcal{G})$. Moreover, due to Theorem 11 we know that $FIN{-}TXT = SMON^d{-}TXT$. Therefore, $\mathcal{L}(\mathcal{G}) \in SMON^d{-}TXT$ implies $\mathcal{L}(\mathcal{G}) \in FIN{-}TXT$. Consequently, there is an IIM M that finitely infers $\mathcal{L}(\mathcal{G})$. On the other hand, $\mathcal{L} \subset \mathcal{L}(\mathcal{G})$. Hence, M finitely infers $\mathcal{L}$, too. Applying Theorem 11 once again yields $\mathcal{L} \in EFIN{-}TXT$. Thus, we conclude $\mathcal{L} \in SMON^d{-}TXT$ which contradicts our assumption. q.e.d.

The situation completely changes, if weak-monotonic learning and its dual counterpart are investigated.

Theorem 15. *For all indexed families $\mathcal{L}$ we have:*

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

Proof. Let $\mathcal{L} \in CWMON^d{-}TXT$. Since $CWMON^d{-}TXT = ELIM{-}TXT$ (cf. Theorem 11), there is an IIM M which $ELIM{-}TXT$–infers $\mathcal{L}$ with respect to the hypothesis space $\mathcal{L}$.

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 in $\mathcal{L}$. Obviously, M can be easily converted

into an IIM $\hat{M}$ which $ELIM{-}TXT$ identifies $\hat{\mathcal{L}}$. Without loss of generality we may assume that $\hat{M}$ is consistent, too (cf. Lange and Zeugmann (1993a)). Furthermore, assume that $\hat{M}$ outputs a canonical number of a singleton language as long as the initial segment presented to $\hat{M}$ does not contain two different strings.

It remains to define an IIM $\tilde{M}$ which dual weak-monotonically infers $\hat{\mathcal{L}}$ with respect to $\hat{\mathcal{L}}$. This proves the above theorem, if we choose $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ such that $(L(G_j))_{j\in\mathbb{N}} = (\hat{L}_j)_{j\in\mathbb{N}}$.

Let $\hat{L}$ be any language in $\hat{\mathcal{L}}$, $t \in \mathit{text}(\hat{L})$, and $x \in \mathbb{N}$. Furthermore, let $j_0, j_1, j_2, \ldots$ denote the sequence of hypotheses generated by $\hat{M}$ when successively fed t. Instead of this sequence, $\tilde{M}$ produces the following sequence of hypotheses $j_0, k_0, j_1, k_1, j_2, k_2, \ldots$ when fed t, too. Thereby, the indices $k_0, k_1, k_2, \ldots$ generated by $\tilde{M}$ as intermediate hypotheses are defined as follows. Let $z \in \mathbb{N}$. If $j_z = j_{z+1}$, then let $k_z = j_z$. Otherwise, i.e., $j_z \neq j_{z+1}$, let k_z denote the number of the singleton language containing the first string in t.

Obviously, $\tilde{M}$ converges to a correct number of $\hat{L}$, since $\hat{M}$ infers $\hat{L}$ when fed t. It remains to show that $\tilde{M}$ works dual weak-monotonically. It suffices to discuss the case $j_z \neq j_{z+1}$. Since $\hat{M}$ exclusively produces consistent hypotheses, it follows $L_{j_z} \supseteq L_{k_z}$ because of the choice of the hypothesis k_z. Hence, $\tilde{M}$ has specialized its former guess L_{j_z}. On the other hand, $j_z \neq j_{z+1}$ implies that at least two different strings occur in the initial segment $\hat{M}$ is fed. Consequently, L_{k_z} is an inconsistent hypothesis and, therefore, the mind change from k_z to j_{z+1} is a justified one. Thereby, $\tilde{M}$ has generalized its former hypothesis k_z, since j_{z+1} is consistent. Thus, we may conclude that $\tilde{M}$ satisfies the dual weak-monotonicity constraint. q.e.d.

In fact, the IIM $\tilde{M}$ defined above realizes a suitable interplay between learning by generalization and learning by specialization. Thereby, $\tilde{M}$ doubles the number of mind changes which $\hat{M}$ performs when processing the same text.

An analogous result can be shown for weak-monotonic learning, too (cf. Section 8, Corollary 40).

Theorem 16. *For all indexed families $\mathcal{L}$ we have:*

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

5.3. Limiting Recursive Compilers

This subsection is devoted to the problem why an indexed family $\mathcal{L}$ that can be learned with respect to some hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ might become non-inferable with respect to other hypothesis spaces $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ satisfying $\mathcal{L} \subseteq \mathcal{L}(\hat{\mathcal{G}})$. A first hint how to answer this question has already been given by Gold (1967). Namely, he proved that, whenever there is a limiting recursive compiler (cf. Definition 9 below) from $\mathcal{G}$ into $\hat{\mathcal{G}}$, then any IIM inferring a class $\mathcal{L}$ of languages with respect to $\mathcal{G}$ can easily be converted into one that learns $\mathcal{L}$ with respect to $\hat{\mathcal{G}}$. Considering indexed families being learnable with respect to some space $\mathcal{G}$ of hypotheses, we could prove that there is always a limiting recursive compiler from $\mathcal{G}$ into $\mathcal{L}$. The same is, *mutatis mutandis*, true for finite learning, i.e., there is always a recursive compiler from $\mathcal{G}$ into

$\mathcal{L}$. However, if some monotonicity requirement is involved, then the situation considerably changes. The reason for that phenomenon is as follows. A limiting recursive compiler in general does not preserve any of the introduced monotonicity demands. But even if it does, it is a highly non-trivial task to convert an IIM that, for example, class preservingly learns an indexed family $\mathcal{L}$ with respect to some appropriate chosen hypothesis space $\mathcal{G}$ into an IIM exactly learning $\mathcal{L}$. The latter difficulty is caused by the fact that one has to combine two limiting processes into one.

For the sake of presentation we give only one of the theorems obtained, since it does already suffice to convey the spirit of the insight achievable. We start with the formal definition of limiting recursive compilers.

Definition 9. *Let $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ and $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ be two spaces of hypotheses such that $L(\mathcal{G}) = \mathcal{L}(\hat{\mathcal{G}})$.* ***A recursive function*** $\mathbf{f}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$ ***is said to be a limiting recursive compiler from $\mathcal{G}$ into $\hat{\mathcal{G}}$*** *iff $k := \lim_{x\to\infty} f(j,x)$ exists and satisfies $L(G_j) = L(\hat{G}_k)$ for all $j\in\mathbb{N}$.*

Next we introduce limiting recursive compilers fulfilling a certain monotonicity demand.

Definition 10. *Let $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ and $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ be two hypothesis spaces such that $\mathcal{L}(\mathcal{G}) = \mathcal{L}(\hat{\mathcal{G}})$.* ***A limiting recursive compiler*** $\mathbf{f}$ ***from $\mathcal{G}$ into $\hat{\mathcal{G}}$ is said to be strong-monotonic*** *iff $L(\hat{G}_{f(j,x)}) \subseteq L(\hat{G}_{f(j,x+1)})$ for all $j,x\in\mathbb{N}$*

Now we are ready to present the announced characterization comparing the power of exact and class preserving learning algorithms under the constraint to learn strong-monotonically. Note that the proof presented below is a considerably improved version of that one in Lange and Zeugmann (1993c).

Theorem 17. *Let $\mathcal{L}$ be an indexed family and let $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ be a hypothesis space such that $\mathcal{L}\in SMON{-}TXT$ with respect to $\mathcal{G}$. Then we have:*

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

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

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

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

Sufficiency. Let $\mathcal{L}\in SMON{-}TXT$ with respect to $\mathcal{G}$ be witnessed by $\hat{M}$, and let f be a strong-monotonic limiting recursive compiler from $\mathcal{G}$ into $\mathcal{L}$. Without loss of generality we may assume that $\hat{M}$ exclusively outputs consistent hypotheses (cf. Lange and Zeugmann (1993a)). We have to define an IIM M that $ESMON{-}TXT$–infers $\mathcal{L}$. The main difficulty we have to deal with is the combination of two limiting recursive processes into one yielding an IIM strong-monotonically inferring $\mathcal{L}$ with respect to $\mathcal{L}$.

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

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

(A) Let $\hat{M}(t_x) = j$. Determine the least $y \in \mathbb{N}$ such that $t_x^+ \subseteq L_{f(j,y)}$. Output $f(j,y)$ and request the next input."

Recall that $\hat{M}$ is consistent. Consequently, $\hat{M}(t_x) = j$ implies $t_x^+ \subseteq L(G_j)$. Since f defines a limiting recursive compiler from $\mathcal{G}$ into $\mathcal{L}$, there exists a $y \in \mathbb{N}$ such that $L_{f(j,y)} = L(G_j)$. Furthermore, since $\mathcal{L}$ is an indexed family, M always effectively finds an index $y \in \mathbb{N}$ satisfying $t_x^+ \subseteq L_{f(j,y)}$. Thus, Instruction (A) terminates and M is indeed an IIM.

Next, we show that M infers L when fed t. Since $\hat{M}$ witnesses $\mathcal{L} \in SMON{-}TXT$, there is a $z \in \mathbb{N}$ such that $\hat{M}(t_{z+r}) = j$ with $L(G_j) = L$ for all $r \in \mathbb{N}$. Since f defines a strong monotonic limiting recursive compiler from $\mathcal{G}$ into $\mathcal{L}$, there is a least $y \in \mathbb{N}$ such that $L_{f(j,y)} = L(G_j)$. Furthermore, $L_{f(j,n)} \subset L_{f(j,y)}$ for all $n < y$. By definition, M converges to the correct guess $L_{f(j,y)}$ when fed t.

Finally, it remains to show that M behaves strong-monotonically when processing the text t for L. First, we prove the following claim.

Claim. $\hat{M}(t_x) = j$ implies $L(G_j) \subseteq \hat{L}$ for all $\hat{L} \in \mathcal{L}$ satisfying $t_x^+ \subseteq \hat{L}$.

Let $\hat{L} \in \mathcal{L}$ such that $t_x^+ \subseteq \hat{L}$. Obviously, $t_x^+ \subseteq \hat{L}$ implies that there is a text $\hat{t}$ for $\hat{L}$ having the initial segment t_x. Since $\hat{M}$, in particular, strong-monotonically infers $\hat{L}$ on $\hat{t}$, $\hat{M}(t_x) = \hat{M}(\hat{t}_x) = j$ implies $L(G_j) \subseteq \hat{L}$. This proves the claim.

Now, let $f(j,y)$ and $f(k,z)$ denote any two different hypotheses subsequently generated by M when fed t, i.e., $M(t_x) = f(j,y)$ and $M(t_{x+r}) = f(k,z)$ for any $x \in \mathbb{N}$ and $r \in \mathbb{N}^+$. It remains to show $L_{f(j,y)} \subseteq L_{f(k,z)}$.

Case 1. $j = k$

By the definition of M, $y < z$. Since f defines a strong-monotonic limiting recursive compiler from $\mathcal{G}$ into $\mathcal{L}$, $y < z$ directly implies $L_{f(j,y)} \subseteq L_{f(j,z)}$.

Case 2. $j \neq k$

By definition of a strong-monotonic limiting recursive compiler $L_{f(j,y)} \subseteq L(G_j)$. Taking M's definition into consideration we obtain $t_{x+r}^+ \subseteq L_{f(k,z)}$. Therefore, $t_x^+ \subseteq L_{f(k,z)}$, too. On the other hand $\hat{M}(t_x) = j$ by the definition of M. Thus, by applying the claim above we obtain $L(G_j) \subseteq L_{f(k,z)}$. Consequently, $L_{f(j,y)} \subseteq L(G_j) \subseteq L_{f(k,z)}$ and, therefore, $L_{f(j,y)} \subseteq L_{f(k,z)}$.

This proves the theorem. q.e.d.

Let us finish our survey on the learnability in dependence on the hypothesis space with the remark that this filed is large and the discourse here is brief. Further information concerning this subject is in part provided in the subsequent sections. Finally, the most intriguing open problems are outlined in Section 9.

6. Iterative IIMs

Within the standard definition of inductive inference machines the limitation of space in realistic computations is not considered. Weakening the requirement that a

learner has always access to the whole initial segment of a text (an informant) it has been fed results in the concept of iterative learning. An iterative IIM is only allowed to use its last guess and the next string of a text and an informant, respectively, in order to produce its next guess. From the viewpoint of potential applications this approach seems to be well-suited. As discussed in Section 3, the class of all pattern languages can be identified by an iterative IIM from text. On the other hand, the question naturally arises whether this restriction seriously affects the learning capabilities. In answering this question, we are mainly interested in estimating the power of iterative learners which are required to fulfill simultaneously certain monotonicity constraints.

6.1. On the Strength of Iterative IIMs

Conceptually, it seems to be appropriate to think about an iterative IIM M as follows: When fed a text (an informant) of a language M is supposed to learn, M defines a sequence $(M_n)_{n\in\mathbb{N}}$ of learning devices each of which takes as its input the output of its predecessor. Furthermore, since iterative learners are always required to produce an output (cf. Definition 8), it seems to be reasonable to consider exclusively iterative IIMs which are allowed to work with respect to a class comprising hypothesis space. On the other hand, we have seen that the learning capabilities of monotonic IIMs essentially depend on the selection of the underlying hypothesis space (cf. Section 5). Consequently, it is interesting to know whether or not iterative learners are also sensitive to the the choice of the underlying space of hypotheses.

Theorem 18. $EIT{-}TXT \subset IT{-}TXT$

Proof. By definition it suffices to show $IT{-}TXT \setminus EIT{-}TXT \neq \emptyset$. Again, we reduce the halting problem to a suitably chosen learning task. The desired indexed family $\mathcal{L}_{eit} = (L_{\langle k,j\rangle})_{k,j\in\mathbb{N}}$ is defined as follows. For all $k \in \mathbb{N}$, we set $L_{\langle k,0\rangle} = \{a^{k+1}\} \cup \{b^n \mid n \in \mathbb{N}^+\}$. For all $j \geq 1$, we distinguish the following cases.

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

We set $L_{\langle k,j\rangle} = \{a^{k+1}\}$.

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

Then, we set $L_{\langle k,j\rangle} = \{a^{k+1}\} \cup \{b^m \mid 1 \leq m \leq x\}$.

Obviously, $\mathcal{L}_{eit}$ is an indexed family. The non-learnability of $\mathcal{L}_{eit}$ in the sense of $EIT{-}TXT$ is due to the following facts. For every $k \in \mathbb{N}$, there is exactly one index for the infinite language $L_{\langle k,0\rangle}$. Moreover, no IIM weak-monotonically infers $\mathcal{L}_{eit}$ with respect to $\mathcal{L}_{eit}$. Hence, every IIM learning $\mathcal{L}_{eit}$ has to produce at least once an overgeneralized hypothesis. Therefore, it has sometimes to shrink its guess to a finite language. But afterwards, it might receive data forcing it to output the corresponding number of the relevant infinite language. Now, every iterative IIM exactly learning $\mathcal{L}_{eit}$ is in serious trouble, since the only available hypothesis does not suffice to memorize the fact that the shrunk guess has been provably rejected. We continue with the formal proof.

Claim 1. $\mathcal{L}_{eit} \notin EIT{-}TXT$

Suppose that there exists any IIM M which $EIT{-}TXT$-identifies $\mathcal{L}_{eit}$. Let us consider M's behavior when fed the text $t = a^{k+1}, b, b^2, \ldots$ for the language $L_{\langle k,0\rangle}$.

Since $\mathcal{L}_{eit} \notin EWMON{-}TXT$ (cf. Lange and Zeugmann (1993b)), there have to be indices $k, x \in \mathbb{N}$ such that $\varphi_k(k)\downarrow$ with $\Phi_k(k) > x$ and M outputs the hypothesis $\langle k, 0\rangle$ after processing t_x. Obviously, t_x serves as an initial segment of a text for $L_{\langle k,\Phi_k(k)\rangle}$, too. Thus, there has to be a string $s \in L_{\langle k,\Phi_k(k)\rangle}$ such that $M(\langle k, 0\rangle, s) = \langle k, y\rangle$ for some $y \geq \Phi_k(k)$. (Note that $L_{\langle k,y\rangle} = L_{\langle k,\Phi_k(k)\rangle}$, if $y \geq \Phi_k(k)$.) On the other hand, M has, in particular, to infer the infinite language $L_{\langle k,0\rangle}$ from its text $\hat{t} = t_x, s, b, s, b^2, \ldots$ Since the string s appears infinitely many times, M outputs infinitely many times the wrong hypothesis $\langle k, y\rangle$. Thus, M fails to converge to a correct guess on $\hat{t}$, a contradiction.

Claim 2. $\mathcal{L}_{eit} \in IT{-}TXT$

We sketch the underlying idea, only. Now, assume any class preserving hypothesis space $\mathcal{G}$ which contains for every $k \in \mathbb{N}$ at least two different indices, say j_k and $\hat{j}_k$, for the infinite language $L_{\langle k,0\rangle}$. Applying this *a priori* knowledge about the underlying hypothesis space, an iterative IIM M is able to handle overgeneralization. Thereby, M may use each of both semantically equivalent hypotheses to represent different stages. Clearly, as long as M is exclusively fed a^{k+1}, it outputs a canonical number of that singleton language. Now we describe how M uses the semantical equivalent hypotheses. The index j_k may be used to encode that M's last guess $L_{\langle k,0\rangle}$ is a possibly overgeneralized hypothesis which may be changed in some subsequent step. M outputs this hypothesis as long as it has no knowledge whether or not $\Phi_k(k)\downarrow$, i.e., as long as it has exclusively seen strings b^z such that $\neg\Phi_k(k) \leq z$. On the other hand, if M has been fed a string b^z satisfying $\Phi_k(k) \leq z$ then it knows for sure that $\Phi_k(k)\downarrow$. After having gained this knowledge, it never outputs j_k. Instead, it either output an index for the corresponding finite language or, in case enough evidence has been presented, the index $\hat{j}_k$ for $L_{\langle k,0\rangle}$. We omit the details. q.e.d.

As the latter proof shows iterative IIMs may successfully handle overgeneralization. Moreover, their ability to solve the subset problem seriously depends on the choice of the relevant hypothesis space. However, it remained open whether or not the power of iterative IIMs increases, if class comprising hypothesis spaces are admissible. But it is known that $CIT{-}TXT \subset CWMON{-}TXT$ (cf. Lange and Zeugmann (1995)). Hence, iterative learning does not achieve the whole power of weak-monotonic IIMs. Nevertheless, the proof technique presented above is powerful enough to compare exact iterative learning and class preserving weak-monotonic inference. Moreover, the following theorems additionally relate the power of $CMON{-}TXT$ and of $CSMON{-}TXT$ to the capabilities of iterative IIMs.

Theorem 19. [1]

(1) $EIT{-}TXT \setminus CSMON{-}TXT \neq \emptyset$

(2) $EIT{-}TXT \setminus CMON{-}TXT \neq \emptyset$

(3) $EIT{-}TXT \setminus WMON{-}TXT \neq \emptyset$

Proof. First, we proof Assertion (2). We define an indexed family $\mathcal{L}$ over the alphabet $\Sigma = \{a, b\}$ as follows. Let $L_0 = \{a\}^+$ and $L_{k,n} = \{a^j \mid 1 \leq j \leq k\} \cup$

[1] Assertion (3) corrects the erroneous statement $EIT{-}TXT \setminus CWMON{-}TXT \neq \emptyset$ in Zeugmann and Lange (1995).

$\{b^k, a^n, b^n\}$. $\mathcal{L} \in EIT{-}TXT$ can be easily verified. On the other hand, $\mathcal{L} \notin CMON{-}TXT$ results from the following observations. Suppose that there is an IIM M which monotonically infers $\mathcal{L}$ with respect to a class comprising hypothesis space $\mathcal{G}$. Since M has to infer L_0 on its lexicographically ordered text t, there is an $x \in \mathbb{N}$ such that $M(t_x) = j$ with $L(G_j) = L_0$. Now, t_x may be extended to become a text $\hat{t}$ for the language $L_{x,x}$ on which M sometimes has to output a correct hypothesis z, say after processing $\hat{t}_{x+r}$. Obviously, $\hat{t}_{x+r}$ forms an initial segment of a text for the language $L_{x,x+1}$. When fed $\hat{t}_{x+r}$, M's first guess j correctly contains the string $a^{x+1} \in L_{x,x+1}$, but a^{x+1} is incorrectly excluded by its subsequent guess z. Thus, M violates the monotonicity constraint when inferring $L_{x,x+1}$ on a text having the initial segment $\hat{t}_{x+r}$.

Since $CSMON{-}TXT \subset CMON{-}TXT$ Assertion (1) follows immediately. Furthermore, since $CMON{-}TXT \,\#\, WMON{-}TXT$, proving Assertion (3) requires a different approach.

Subsequently, we use the following shorthands. For all $n, m \in \mathbb{N}$, let $\hat{L}_{\langle n,0\rangle} = \{b^n c^j \mid 1 \leq j\}$ and $\hat{L}_{\langle n,m\rangle} = \{b^n c^j \mid 1 \leq j \leq m\}$. The desired indexed family $\mathcal{L} = (L_{\langle k,j\rangle})_{k,j\in\mathbb{N}}$ will be defined as follows. Let $k \in \mathbb{N}$. We distinguish the following cases.

Case 1. $j \leq 1$

Then, we set $L_{\langle k,j\rangle} = \{a^{k+1}\} \cup (\bigcup_{n\in\mathbb{N}} \hat{L}_{\langle n,0\rangle})$.

Case 2. $j \geq 2$

We distinguish the following subcases:

Subcase 2.1. $\neg\, \Phi_k(k) \leq j-1$

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

Subcase 2.2. $\Phi_k(k) \leq j-1 \leq 2\Phi_k(k)$

Let $d = (j-1) - \Phi_k(k)$. Then, we set $L_{\langle k,j\rangle} = \{a^{k+1}\} \cup (\bigcup_{n\leq d} \hat{L}_{\langle n,0\rangle}) \cup (\bigcup_{n>d} \hat{L}_{\langle n,\Phi_k(k)\rangle})$.

Subcase 2.3. $j-1 > 2\Phi_k(k)$

Then, we set $L_{\langle k,j\rangle} = \{a^{k+1}\} \cup \hat{L}_{\langle 0,0\rangle} \cup (\bigcup_{n>0} \hat{L}_{\langle n,\Phi_k(k)\rangle})$.

$\mathcal{L} \notin WMON{-}TXT$ follows by applying our standard proof idea, namely by reducing the halting problem to $\mathcal{L} \in WMON{-}TXT$ (cf. Lange and Zeugmann (1993c)). Next, we define an iterative IIM M which infers $\mathcal{L}$ with respect to the hypothesis space $\mathcal{L}$. Let $L \in \mathcal{L}$ and let $t = s_0, s_1, s_2, \ldots$ be any text for L. Without loss of generality, we may assume that $s_0 = a^{k+1}$ for some $k \in \mathbb{N}$. (If $s_0 \neq a^{k+1}$, M simply ignores all strings presented until a string a^{k+1} appears for the first time.) The IIM M is defined in stages, where Stage x conceptually describes M_x.

Stage 0: Let $s_0 = a^{k+1}$ for some $k \in \mathbb{N}$. Set $j_0 = \langle k, 0\rangle$. Output $\langle k, 0\rangle$, and goto Stage 1.

Stage x: M receives as input j_{x-1} and the $x+1$st element s_x of t. If $s_x = a^{k+1}$ for some $k \in \mathbb{N}$, then set $j_x = j_{x-1}$. Output j_x, and goto Stage $x+1$. Otherwise, $s_x = b^n c^m$ for some $n, m \in \mathbb{N}$.

Case 1. $j_{x-1} = \langle k, 0\rangle$ for some $k \in \mathbb{N}$

Test whether or not $\Phi_k(k) \leq m$. In case it is, set $j_x = \langle k, \Phi_k(k)+1\rangle$, output j_x, and goto Stage $x+1$. Otherwise, set $j_x = j_{x-1}$, output j_x, and goto Stage $x+1$.

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

Set $j_x = j_{x-1}$, output j_x, and goto Stage $x+1$.

Case 3. $j_{x-1} = \langle k, z\rangle$ for some $k \in \mathbb{N}$ and some $z \geq 1$

Test whether or not $s_x = b^n c^m \in L_{\langle k,z\rangle}$. In case it is, set $j_x = j_{x-1}$, output j_x, and goto Stage $x+1$. Otherwise, execute Instruction (A).

(A) Test whether or not $n \leq \Phi_k(k)$. In case it is, set $j_x = \langle k, \Phi_k(k)+n+1\rangle$, output j_x, and goto Stage $x+1$. Otherwise, set $j_x = \langle k, 1\rangle$, output j_x, and goto Stage $x+1$.

It remains to show that M infers $\mathcal{L}$. Let $k, z \in \mathbb{N}$. Assume that a text t for the target language $L = L_{\langle k,z\rangle}$ is presented. If $\varphi_k(k)$ is undefined, then M outputs in every step the guess $\langle k, 0\rangle$. Since $L_{\langle k,0\rangle} = L_{\langle k,z\rangle}$ for all $z \in \mathbb{N}$, M infers L from text t. It remains to discuss the case that $\varphi_k(k)$ is defined.

Because of the definition of $\mathcal{L}$, there has to be an $x \in \mathbb{N}$ such that $s_x = c^m$ with $m \geq \Phi_k(k)$. Hence, M eventually rejects its current guess $\langle k, 0\rangle$ and changes its mind to the guess $\langle k, \Phi_k(k)+1\rangle$. Afterwards, M realizes the subset principle. In particular, M avoids overgeneralization in every subsequent step. Since M has to distinguish between finitely many possible hypotheses, only, M converges to a correct hypotheses. Note that every language is uniquely characterized by infinitely many strings of the form $b^n c^m$ satisfying $m > \Phi_k(k)$ whereas the maximal index n is referring to a correct number of the target language $\mathcal{L}$.

Thus, M behaves as required. This finishes the proof of Assertion (3). q.e.d.

If language learning from informant is investigated, the situation slightly changes. The difference is caused by the fact that the weak-monotonicity constraint does not further restrict learning power. A detailed proof of the theorem below may be found in Lange and Zeugmann (1992).

Theorem 20.

(1) $EIT{-}INF \setminus CSMON{-}INF \neq \emptyset$

(2) $EIT{-}INF \setminus CMON{-}INF \neq \emptyset$

(3) $CIT{-}INF \subset EWMON{-}INF$

6.2. Monotonic Inference by Iterative Machines

In this subsection we aim to give some more insight concerning the trade-offs between information presentation and monotonicity constraints. Our treatment is based on the following perspective. Obviously, an iterative learner has only a limited access to the history of the learning process. Contrary to that, a monotonic learner can inspect the whole initial segment it has been fed. Thus, it may recompute the whole sequence of hypotheses created so far, and may incorporate this knowledge into

the production of its actual guess. On the other hand, learning under monotonicity constraints provides additional *a priory* knowledge concerning the relation of subsequently produced hypotheses. Hence, it is only natural to ask whether or not this knowledge suffices to learn iteratively, too. As we will see, the answer to this question heavily depends on the monotonicity constraint involved as well as on the class of admissible hypothesis spaces.

Theorem 21. $EMON{-}TXT \setminus CIT{-}TXT \neq \emptyset$

Proof. Let $L_1 = \{a\}^*$ and $L_{k,n} = \{a^z \mid z \leq k\} \cup \{a^z \mid z \geq n\} \cup \{b^k, b^n\} \cup \{c\}$ for all $k, n \in \mathbb{N},\ k, n > 1$ and $k + 2 < n$. Finally, we set $L_{k,n,m} = L_{k,n} \setminus \{c\} \cup \{a^m\}$ for all $k, n \in N$ as above and $k < m < n$. Then define $\mathcal{L}$ to be the collection of all $L_1,\ L_{k,n}$ and $L_{k,n,m}$. It is not hard to prove that $\mathcal{L} \in EMON{-}TXT$. The following weakness of iterative learners will be exploited to prove $\mathcal{L} \notin CIT{-}TXT$. Suppose to the contrary that an iterative IIM M yields successful inference. Let t be any text for L_1. Since M has to infer L_1, it has to reach a point of stabilization, i.e., after this point M has to repeat its last guess, no matter which string $s \in L_1$ is actually presented. Let a^k be the longest string M has fed before reaching the point of stabilization. Later on, M has no chance to deduce from its last hypothesis whether or not a particular string $s \in L_1$ has been fed to it. But this is necessary to distinguish, for instance, between the languages $L_{k,k+3,k+1}$ and $L_{k,k+3,k+2}$. Hence, M is fooled. q.e.d.

The latter theorem directly yields the following consequence. Monotonic as well as weak-monotonic inference with iterative IIMs is less powerful than with ordinary machines.

Corollary 22. *Let $\lambda \in \{E, \varepsilon, C\}$. Then we have:*

(1) $\lambda MON{-}IT{-}TXT \subset \lambda MON{-}TXT$

(2) $\lambda WMON{-}IT{-}TXT \subset \lambda WMON{-}TXT$

Next, we study strong-monotonic inference performed by iterative IIMs. As our next theorem shows, there is a peculiarity. Namely, class *comprising strong-monotonic* inference from positive data is precisely as powerful as *class comprising strong-monotonic and iterative* learning. Moreover, class comprising hypothesis spaces are inevitable to achieve this equality.

Theorem 23.

(1) $ESMON{-}IT{-}TXT \subset ESMON{-}TXT$

(2) $SMON{-}IT{-}TXT \subset SMON{-}TXT$

(3) $CSMON{-}IT{-}TXT = CSMON{-}TXT$

Proof. First, we show (1) and (2). It suffices to present an indexed family $\mathcal{L} = (L_j)_{j \in \mathbb{N}}$ witnessing $ESMON{-}TXT \setminus SMON{-}IT{-}TXT \neq \emptyset$. Let $\mathcal{L}$ denote any canonical enumeration of all finite languages over $\Sigma = \{a\}$ which does not contain the singleton languages $\{a\}$ and $\{aa\}$.

Obviously, $\mathcal{L} \in ESMON{-}TXT$. The main reason for $\mathcal{L} \notin SMON{-}IT{-}TXT$ can be described as follows. Suppose, M is initially fed the string a. Then, M may either

remain in its initial stage or it may guess a language L from $\mathcal{L}$. However, the latter cannot be done without violating the strong-monotonicity constraint, since $L \neq \{a\}$ for all $L \in \mathcal{L}$. The same argument applies when M is initially fed the string aa. Hence, in both cases M is forced to maintain its initial stage. But this implies that M guesses the same sequence of hypotheses when successively fed $t = a, a^3, a^3, \ldots$ and $\hat{t} = aa, a^3, a^3, \ldots$, respectively. Consequently, M fails to infer at least one of the corresponding finite languages $L = \{a, a^3\}$ and $\hat{L} = \{aa, a^3\}$ in the sense of $SMON\text{-}IT\text{-}TXT$, respectively.

Next, we prove Assertion (3). Let M strong-monotonically infer $\mathcal{L}$ with respect to a class preserving space of hypotheses $\mathcal{G}$. Without loss of generality, we may assume that M outputs in every step a consistent hypothesis (cf. Lange and Zeugmann (1993a)). First, we define another strong-monotonic IIM $\hat{M}$ which infers $\mathcal{L}$. In doing so, we choose the hypothesis space $\hat{\mathcal{G}}$ which is obtained from $\mathcal{G}$ by enumerating its closure with respect to finite unions. Now, let $L \in \mathcal{L}$ and $t = s_0, s_1, \ldots$ be any text for L.

$\hat{M}(t_x) =$ "If $x = 0$, compute $M(t_0) = j$. Output, the canonical index $\hat{\jmath}_0$ of $L(G_j)$ in $\hat{\mathcal{G}}$. Otherwise, execute Instruction (A).

(A) Let $\hat{\jmath}_k = \hat{M}(t_x)$. If $s_x \subseteq L(\hat{G}_{\hat{\jmath}_k})$, repeat the hypothesis $\hat{\jmath}_k$. Otherwise, goto (B).

(B) Compute the initial segment $\hat{t}_z$ of the lexicographically ordered text of $L(\hat{G}_{\hat{\jmath}_k})$ which contains all strings being smaller than s_x with respect to the underlying lexicographical ordering. Compute $j = M(\hat{t}_z, s_x)$. Output the canonical index of the language $L(\hat{G}_{\hat{\jmath}_k}) \cup L(G_j)$ in $\hat{\mathcal{G}}$."

By definition, $L(\hat{G}_{\hat{\jmath}_k}) \subseteq L(\hat{G}_{\hat{\jmath}_{k+1}})$ for every $k \in \mathbb{N}$. Thus, $\hat{M}$ works strong-monotonically, too. Moreover, $\hat{M}$ exclusively produces consistent hypotheses, because M is a consistent IIM. When fed *any* initial segment t_x of *any* text for L, M always outputs a guess j such that $L(G_j) \subseteq L$. Obviously, the same statement is true, if t_x is an initial segment of a text for *any* of L's sublanguages. Now, taking $\hat{M}'s$ definition into account it can be easily verified that $\hat{M}$ never outputs an overgeneralized hypothesis.

It remains to show that $\hat{M}$ infers L from text t. Since $\hat{M}$ exclusively performs justified mind changes, it suffices to show that $\hat{M}$ outputs once a correct hypothesis. This happens trivially, if L is a finite language, since $\hat{M}$ is a consistent IIM. Otherwise, let L be infinite. In order to verify that $\hat{M}$ succesfully handles this case, too, one has to take into consideration that M, in particular, strong-monotonically infers L from its lexicographically ordered text t^L. (Note that this assumes that L is infinite.) Thus, $M(t_x^L) = j$ with $L(G_j) = L$ for some $x \in \mathbb{N}$. On the other hand, t is a text for L. Hence, there is a least $y \in \mathbb{N}$ such that $range(t_x^L) \subseteq t_y^+$. Moreover, $\hat{M}(t_y) = \hat{\jmath}_y$ with $range(t_x^L) \subseteq L(\hat{G}_{\hat{\jmath}_y})$. If $L = L(\hat{G}_{\hat{\jmath}_y})$, we are done. Otherwise, $L(\hat{G}_{\hat{\jmath}_y}) \subset L$. Then, there exists a least $r \in \mathbb{N}$ such that $s_{y+r} \notin range(t_x^L)$. Since $range(t_x^L) \subseteq L(\hat{G}_{\hat{\jmath}_y}) \subset L$, the string s_{y+r} forces $\hat{M}$ to compute an extension of L's lexicographically ordered text having the initial segment t_x^L. Taking again $\hat{M}$'s definition into account it follows that $\hat{M}(t_{y+r}) = \hat{\jmath}_{y+r}$ with $L(G_j) \subseteq L(\hat{G}_{\hat{\jmath}_{y+r}})$. Since $\hat{M}$ avoids overgeneralization, we are done.

Finally, a closer look to the definition of $\hat{M}$ shows that, when fed t_x, $\hat{M}$'s output depends only on its last guess $\hat{j}_{x-1}$ and the actual string s_x. Consequently, $\hat{M}$ witnesses $\mathcal{L} \in CSMON{-}IT{-}TXT$. q.e.d.

Finally, we summarize the corresponding results concerning monotonic inference from positive and negative examples (cf. Lange and Zeugmann (1992)).

Theorem 24. *Let $\lambda \in \{E, \varepsilon, C\}$. Then we have:*

(1) $\lambda MON{-}IT{-}INF \subset \lambda MON{-}INF$

(2) $\lambda WMON{-}IT{-}INF \subset \lambda WMON{-}INF$

Furthermore, concerning strong-monotonic learning the following theorem can be proved. Thereby, the same idea already used in the demonstration of Theorem 23 applies *mutatis mutandis*.

Theorem 25.

(1) $ESMON{-}IT{-}INF \subset ESMON{-}INF$

(2) $SMON{-}IT{-}INF \subset SMON{-}INF$

Up to now, it remains open whether class comprising strong-monotonic inference from informant can be performed by iterative IIMs without limiting learning power.

7. Trading Monotonicity Constraints Versus Efficiency

This section deals with the efficiency of learning. The measure of efficiency we use is the number of mind changes an IIM is allowed to perform. Starting with the pioneering paper by Barzdin and Freivalds (1972) this measure of efficiency has been intensively studied (cf., e.g., Barzdin, Kinber and Podnieks (1974), Barzdin and Freivalds (1974), Case and Smith (1983), Wiehagen, Freivalds and Kinber (1984)). 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. Therefore, we continue with a short survey that comprises relevant results concerning the inferability of indexed families. Mukouchi (1992) considered exact learning from both, text and informant, and established the following hierarchies:

$$ELIM_0{-}TXT \subset ELIM_1{-}TXT \subset ... \subset \bigcup_{n\in\mathbb{N}} ELIM_n{-}TXT \subset ELIM_*{-}TXT$$

$$ELIM_0{-}INF \subset ELIM_1{-}INF \subset ... \subset \bigcup_{n\in\mathbb{N}} ELIM_n{-}INF \subset ELIM_*{-}INF$$

Subsequently, in Lange and Zeugmann (1993b) we extended the latter result to the class preserving case. Moreover, we considered the problem whether or not information presentation can always be traded versus efficiency, and obtained $ELIM_{n+1}{-}TXT \setminus LIM_n{-}INF \neq \emptyset$ for all $n \in \mathbb{N}$ as well as $ELIM_1{-}INF \setminus LIM{-}TXT \neq \emptyset$. Furthermore, we studied the influence of an appropriate choice of the hypothesis space

on the efficiency of learning. As it turned out, at least one mind change can be saved provided the right hypothesis space is used. Recently, Lange (1994) sharpened the latter result in the strongest possible way for learning from text as well as from informant, thereby, handling class comprising learning, too. Finally, we additionally succeeded to characterize $LIM_n{-}TXT$ and $LIM_n{-}INF$ in terms of uniformly generable recursive finite tell-tales (cf. Lange and Zeugmann (1993b) as well as Theorem 3) and M. Sato extended this result to exact learning (cf. Mukouchi (1994)).

Discussing some of the results outlined above, Kinber (1992) proposed the following interesting problem.

Does any of the monotonicity constraints defined in Section 2 influence the efficiency of learning?

Clearly, this question is directly related to the problem how 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 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 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. A partial answer to this problem can already be found in Lange, Zeugmann and Kapur (1992). There it has been shown that $CWMON^d{-}TXT = CLIM{-}TXT$. However, the construction presented uniformly transforms any IIM that learns a target indexed family in the limit into one that fulfills the dual weak-monotonicity constraint. But the price paid is high. The dual weak-monotonic learner may be forced to change its mind twice as often than the original IIM (cf. Section 5, Theorem 15). It is open whether or not this bound is tight. Furthermore, Lange, Zeugmann and Kapur (1992) presents results showing that indexed families learnable with an *a priori* fixed number of mind changes under some monotonicity constraint can become non-inferable at all if the monotonicity demand is strengthened.

Our approach below deals with a problem of higher granularity. We always start with a target indexed family inferable under some 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 monotonicity requirement might help to uniformly reduce the number of mind changes. As we shall see, there is no unique answer to this problem. Finally, in the following we restrict ourselves to consider learning from positive data and incorporate the number of mind changes into the definition of all types of monotonic learning in the same way as it has been done in Definition 3. The resulting learning types are denoted by $\lambda SMON_n{-}TXT$, $\lambda MON_n{-}TXT$ and $\lambda WMON_n{-}TXT$, where $n \in \mathbb{N} \cup \{*\}$, and $\lambda \in \{C, \varepsilon, E\}$.

7.1. Strong-Monotonic Inference

We start our investigations with the strongest possible monotonicity constraint, i.e., with $SMON{-}TXT$ and its variations. Note that it does not make sense to consider $ESMON^d{-}TXT$ or $SMON^d{-}TXT$, since $SMON^d{-}TXT = EFIN{-}TXT$, and hence, there is nothing to speed-up. 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.

Theorem 26. *Let $\mathcal{L}$ be an indexed family. Then, for every $n \in \mathbb{N}^+$ we have:*

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

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

Proof. The proof is based on the following observations.

(A) Any strong-monotonically working IIM $\hat{M}$ can be simulated by a consistent, conservative, and strong-monotonic IIM M that performs at most as many mind changes than $\hat{M}$ does (cf. Lange and Zeugmann (1993a)).

(B) Let $\mathcal{L}$ be any indexed family with $\mathcal{L} \in ESMON_{n+1}\text{–}TXT \setminus ESMON_n\text{–}TXT$. Furthermore, let $\mathcal{L} \in ESMON_{n+1}\text{–}TXT$ be witnessed by M, where M is chosen in accordance with (A). Since $\mathcal{L} \notin ESMON_n\text{–}TXT$, there has to be an $L \in \mathcal{L}$ 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 and conservative, we directly obtain that $L_{j_0} \subset \cdots \subset L_{j_{n+1}} = L$.

Now, $\mathcal{L} \notin CLIM_n\text{–}TXT$ is a direct consequence of Proposition 3.7 by Mukouchi (1994). Applying the same arguments, one easily proves Assertion (2). q.e.d.

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 $ESMON_{n+1}\text{–}TXT$ ($SMON_{n+1}\text{–}TXT$), but cannot be class preservingly and strong-monotonically learned with at most n mind changes for some $n \in \mathbb{N}$.

Next we consider the class comprising case. Interestingly enough, now the topological argument used above does not apply any more. 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 27. *For every $n \in \mathbb{N}^+$ there exists an indexed family $\mathcal{L}$ such that*

(1) $\mathcal{L} \in CSMON_{n+1}\text{–}TXT \setminus CSMON_n\text{–}TXT$,

(2) $\mathcal{L} \in ELIM_n\text{–}TXT$.

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. Let $k, j \in \mathbb{N}$. We set $L_{3\langle k,j\rangle} = \{a^k b^z \mid z \in \mathbb{N}^+\}$. The remaining languages will be defined as follows.

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

Then we set $L_{3\langle k,j\rangle+1} = L_{3\langle k,j\rangle+2} = L_{3\langle k,0\rangle}$.

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

Let $m = \Phi_k(k)$. Now we set $L_{3\langle k,j\rangle+1} = \{a^k b^z \mid 1 \leq z \leq m\} \cup \{a^k c^m\}$, and $L_{3\langle k,j\rangle+2} = L_{3\langle k,0\rangle} \cup \{a^k d^m\}$.

Since the predicate "$\Phi_i(x) = y$" is uniformly decidable for all $i, x, y \in \mathbb{N}$, it is easy to see that $\mathcal{L} = (L_z)_{z\in\mathbb{N}}$ is an indexed family. Whenever $\Phi_k(k)\downarrow$, the main problem for any strong-monotonic IIM consists in learning the finite language $L_{3\langle k,\Phi_k(k)\rangle+1}$ with at most one mind change. Hence, for proving $\mathcal{L} \in CSMON_2{-}TXT$, another ingredient is required, i.e., a suitable choice of a hypothesis space (cf. Claim A). The harder part is to show that $\mathcal{L} \notin CSMON_1{-}TXT$. 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 C).

We continue with the formal proof.

Claim A. $\mathcal{L} \in CSMON_2{-}TXT$.

First of all we define a suitable hypothesis space $\tilde{\mathcal{L}} = (\tilde{L}_i)_{i\in\mathbb{N}}$. For all $k, j \in \mathbb{N}$ and $z \in \{0, 1, 2\}$, we set:

$$\tilde{L}_{3\langle k,j\rangle+z} = \begin{cases} \bigcap_{j\in\mathbb{N}} L_{3\langle k,j\rangle+1}, & \text{if } j = 0, \\ L_{3\langle k,j-1\rangle+z}, & \text{otherwise.} \end{cases}$$

It is not hard to see that $\tilde{\mathcal{L}}$ is indeed an indexed family. Now we define an IIM M which strong-monotonically identifies $\mathcal{L}$ with respect to $\tilde{\mathcal{L}}$.

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

$M(t_x) =$ "Determine the unique k such that $a^k b^z \in t_x^+$ for some $z \in \mathbb{N}$. Test whether or not $t_x^+ \subseteq \tilde{L}_{3\langle k,0\rangle}$. In case it is, output $3\langle k, 0\rangle$. Otherwise, goto (A).

(A) Compute $m = max\{z \mid a^k b^z \in \tilde{L}_{3\langle k,0\rangle})\}$. In case that $a^k c^m \in t_x^+$, output $3\langle k, m\rangle + 1$. Otherwise, goto (B).

(B) If $a^k d^m \in t_x^+$, then output $3\langle k, m\rangle + 2$. Else, output $3\langle k, 1\rangle$."

Now, one straightforwardly verifies that M $CSMON_2$–TXT–learns $\mathcal{L}$. This proves Claim A.

Before showing the second part of Assertion(1), we prove Assertion(2).

Claim B. $\mathcal{L} \in ELIM_1{-}TXT$.

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

$M(t_x) =$ "Determine the unique k such that $a^k b^z \in t_x^+$ for some $z \in \mathbb{N}$. Test whether or not $t_x^+ \subseteq L_{\langle k,0\rangle}$. In case it is, output $3\langle k, 0\rangle$. Otherwise, goto (A).

(A) Compute $m = \Phi_k(k)$. In case that $a^k c^m \in t_x^+$, output $3\langle k, m\rangle + 1$. Otherwise, output $3\langle k, m\rangle + 2$."

It remains to show that M infers $\mathcal{L}$ in the sense of $ELIM_1{-}TXT$. 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, $t_x^+ \notin L_{3\langle k,0\rangle}$ 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 $ELIM_1{-}TXT$–infers $\mathcal{L}$. This proves Claim B.

Claim C. $\mathcal{L} \notin CSMON_1{-}TXT$.

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

Procedure HALT

"Let $k \in \mathbb{N}$, and let t be the canonical text for $L_{3\langle k,0\rangle}$. For $x = 0, 1, ...,$ 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 **HALT** is an effective procedure. In particular, M has to infer $L_{3\langle k,0\rangle}$ from t. Hence, there is a z such that M on input t_z computes a hypothesis j. Hence, **HALT** is recursive and terminates for all $k \in \mathbb{N}$.

It remains to show that **HALT** correctly works. Obviously, if the output is $\varphi_k(k)\downarrow$, then $\varphi_k(k)$ is indeed defined. Suppose, **HALT** 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 $L(G_j) \notin \mathcal{L}$. Furthermore, M has to infer $L_{3\langle k,0\rangle}$ from its canonical text. Hence, there has to be an $y > z$ such that $M(t_y) = r$ and $L(G_r) = L_{3\langle k,0\rangle}$. Therefore, M performs at least one mind change when seeing t_y. Finally, due to our construction, there is a language $L' \in \mathcal{L}$ such that $t_y^+ \subseteq L'$ and $L' \neq L_{3\langle k,0\rangle}$, namely $L' = L_{3\langle k,0\rangle} \cup \{a^k d^m\}$. Consequently, t_y may be extended to a text for L' 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 (1993b) (cf. proof of Theorem 11). q.e.d.

At this point it is only natural to ask whether the latter theorem generalizes to all indexed families from $CSMON_{n+1}{-}TXT \setminus CSMON_n{-}TXT$ not belonging to $SMON{-}TXT$. As we show, removing the requirement to learn strong-monotonically does not necessarily lead to a speed-up with respect to the number of mind changes.

Theorem 28. *For all $n \in \mathbb{N}$, there exists an indexed family $\mathcal{L}$ such that*

(1) $\mathcal{L} \in CSMON_{n+1}{-}TXT \setminus CSMON_n{-}TXT$,

(2) $\mathcal{L} \notin SMON{-}TXT$,

(3) $\mathcal{L} \notin ELIM_n{-}TXT$.

Proof. We consider the following indexed family $\mathcal{L} = (L_k)_{k\in\mathbb{N}}$. 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 $\Phi_k(k)$ diverges. Then it is easy to show that $\mathcal{L} \in CSMON_1{-}TXT \setminus SMON{-}TXT$. On the other hand, one straightforwardly verifies that $\mathcal{L}$ cannot be finitely inferred. Again, an easy application of the same "lifting" technique as in Lange and Zeugmann (1993b) directly yields the Theorem for all $n \in \mathbb{N}$. q.e.d.

Theorem 28 directly yields the problem whether or not Theorem 27 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 29. *For every $n \in \mathbb{N}^+$ there exists an indexed family $\mathcal{L}$ such that*

(1) $\mathcal{L} \in CSMON_{n+1}{-}TXT \setminus CSMON_n{-}TXT$,

(2) $\mathcal{L} \in EMON_1{-}TXT$.

Proof. We restrict ourselves to present the case $n = 2$, since it suffices to explain the *proof technique* developed. The main idea is to suitably iterate the proof technique presented in the demonstration of Theorem 27. Therefore, we incorporate one more halting problem into the definition of the indexed family $\mathcal{L}$ witnessing $\mathcal{L} \in CSMON_3{-}TXT \setminus CSMON_2{-}TXT$, and $\mathcal{L} \in ELIM_1{-}TXT$. 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:

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

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

Then we set $L_{4\langle k_1,k_2,j\rangle+1} = L_{4\langle k_1,k_2,j\rangle+2} = L_{4\langle k_1,k_2,j\rangle+3} = L_{4\langle k_1,k_2,0\rangle}$.

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

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

Furthermore, we distinguish the following subcases.

Subcase 2.1. $\neg\Phi_{k_2}(k_2) \leq j$

Then let $L_{4\langle k_1,k_2,j\rangle+2} = L_{4\langle k_1,k_2,j\rangle+3} = L_{4\langle k_1,k_2,0\rangle}$.

Subcase 2.2. $\Phi_{k_2}(k_2) \leq j$

Let $\ell = \Phi_{k_1}(k_1)$, and $m = \Phi_{k_2}(k_2)$.
We set $L_{4\langle k_1,k_2,j\rangle+2} = \{a^{\langle k_1,k_2\rangle} b^z \mid 1 \leq z \leq \ell + m\} \cup \{a^{\langle k_1,k_2\rangle} d^{\ell+m}\}$, and
$L_{4\langle k_1,k_2,j\rangle+3} = L_{4\langle k_1,k_2,0\rangle} \cup \{a^{\langle k_1,k_2\rangle} e^{\ell+m}\}$.

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. This is done by the following lemmata.

***Lemma* 1**. $\mathcal{L} \in EMON_1{-}TXT$.

An IIM M witnessing $\mathcal{L} \in ELIM_1{-}TXT$ can be easily defined. Initially, it outputs $4\langle k_1, k_2, 0\rangle$. As long as this guess is consistent, it is repeated. Otherwise, M reads one of the following strings $a^{\langle k_1,k_2\rangle} c^\ell$, $a^{\langle k_1,k_2\rangle} d^{\ell+m}$ or $a^{\langle k_1,k_2\rangle} e^{\ell+m}$. These strings serve as a label as the definition of $\mathcal{L}$ shows. Therefore, M can change its mind to a correct

hypothesis which it repeats subsequently. Moreover, it is easy to see that the possible mind change satisfies the monotonicity requirement. This proves the lemma.

Lemma 2. $\mathcal{L} \in CSMON_3{-}TXT$.

The wanted hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ is defined to be a canonical enumeration of all languages of $\mathcal{L}$ and all languages $\hat{L}_{4\langle k_1,k_2\rangle+1} = \bigcap_{j\in\mathbb{N}} L_{4\langle k_1,k_2,j\rangle+1}$ as well as $\hat{L}_{4\langle k_1,k_2\rangle+2} = \bigcap_{j\in\mathbb{N}} L_{4\langle k_1,k_2,j\rangle+2}$ for all $k_1, k_2 \in \mathbb{N}$. We suppress the technicalities and refer to hypotheses in $\mathcal{G}$ as to canonical numbers of the corresponding languages.

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

$M(t_x) =$ "If $a^{\langle k_1,k_2\rangle}c^{\ell} \subseteq t_x^+$ output the canonical number of $L_{4\langle k_1,k_2,\ell\rangle+1}$.
If $a^{\langle k_1,k_2\rangle}d^{\ell+m} \subseteq t_x^+$ output the canonical number of $L_{4\langle k_1,k_2,\ell+m\rangle+2}$.
If $a^{\langle k_1,k_2\rangle}e^{\ell+m} \subseteq t_x^+$ output the canonical number of $L_{4\langle k_1,k_2,\ell+m\rangle+3}$.
Otherwise, i.e., if $t_x^+ \subseteq \{a^{\langle k_1,k_2\rangle}b^z \mid z \in \mathbb{N}^+\}$, test successively whether or not the canonical number for $\hat{L}_{4\langle k_1,k_2\rangle+1}$, or $\hat{L}_{4\langle k_1,k_2\rangle+2}$ or $L_{4\langle k_1,k_2,0\rangle}$ is consistent. Output the first consistent hypothesis."

The definitions of $\mathcal{L}$ and $\mathcal{G}$ directly imply that M satisfies the strong-monotonicity constraint. Moreover, it is easy to see that M learns $\mathcal{L}$ with respect to $\mathcal{G}$. Hence, it remains to show that three mind changes are sufficient. Obviously, the worst case occurs when M is forced to output successively the canonical numbers for $\hat{L}_{4\langle k_1,k_2\rangle+1}$, $\hat{L}_{4\langle k_1,k_2\rangle+2}$ and $L_{4\langle k_1,k_2,0\rangle}$ before seeing $a^{\langle k_1,k_2\rangle}e^{\ell+m}$. However, even in this case M performs precisely three mind changes. This proves the lemma.

The remaining part, i.e., $\mathcal{L} \notin CSMON_2{-}TXT$, is much harder to prove. The critical part is to show that any IIM which strong-monotonically infers $\mathcal{L}$ has to be at least as careful, when fed a text for $L_{4\langle k_1,k_2,0\rangle}$, as the IIM M provided in Lemma 2. For that purpose we need some additional insight into the behavior of every IIM that strong-monotonically learns $\mathcal{L}$. In particular, we are mainly interested in knowing how every IIM inferring $\mathcal{L}$ strong-monotonically behaves when successively fed the lexicographically ordered text for $L_{4\langle k_1,k_2,0\rangle}$. The desired information is provided by the following lemma.

Lemma 3. *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 $\mathcal{L} \in CSMON{-}TXT$ 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 $L_{4\langle k_1,k_2,0\rangle}$.

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 $\Phi_{k_1}(k_1)\uparrow$.

Assuming the latter statement we have the following claim.

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

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

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

(A1) Generate successively the lexicographically ordered text t of $L_{4\langle k_1,k_2,0\rangle}$ 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 $L_{4\langle k_1,k_2,0\rangle}$ 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, the lemma follows.

Lemma 4. $\mathcal{L} \notin CSMON_2{-}TXT$.

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

***Lemma* A**. *Given any hypothesis space $\mathcal{G}=(G_j)_{j\in\mathbb{N}}$ and any program for M witnessing $\mathcal{L}\in CSMON_2{-}TXT$, one can effectively construct an algorithm deciding whether or not "$\varphi_{k_2}(k_2)\downarrow$."*

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

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

(A1) For $z=0,\ 1,\ 2, \ldots$ successively compute the lexicographically ordered texts $t^{j_0},\ t^{j_1},\ t^{j_2}, \ldots$ for $L_{4\langle j_0,k_2,0\rangle},\ L_{4\langle j_1,k_2,0\rangle}, \ldots, L_{4\langle j_z,k_2,0\rangle}$ 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, the execution of (A1) has to terminate *)

(A2) Let $f =_{df} \langle j_r, k_2\rangle$ and $\ell=\Phi_{j_r}(j_r)$. Furthermore, we define $\hat{t}_{\ell+y}$ as follows:
$\hat{t}_{\ell+y} = \underbrace{a^f b,\ \ldots,\ a^f b^{x+1},\ \ldots,\ a^f b^{\ell}}_{=t^{j_r}_{\ell-1}},\ a^f b^{\ell},\ \underbrace{a^f b^{\ell+1},\ \ldots,\ a^f b^{\ell+y}}_{y-\text{strings}}$

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

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

($\beta 2$) Compute $j_{\ell+y} = M(\hat{t}_{\ell+y})$.
($\beta 3$) $\Phi_{k_2}(k_2) \leq \ell + y$ is verified. Then output "$\varphi_{k_2}(k_2)\downarrow$."
($\beta 4$) In ($\beta 2$) 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 $\mathcal{A}$ terminates on every input, and behaves correctly.

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

As we have already mentioned, by Lemma 3 we know that the execution of (A1) has to terminate. Hence, it suffices to show that either ($\beta 3$) or ($\beta 4$) happens. Suppose, ($\beta 3$) does not happen. Then, for all $y \in \mathbb{N}$ we have $\neg\Phi_{k_2}(k_2) \leq \ell + y$. Consequently, $\Phi_{k_2}(k_2)\uparrow$. Therefore, when y tends to infinite, then $\hat{t}_{\ell+y}$ converges to a text for $L_{4\langle j_n,k_2,0\rangle}$, 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 the claim.

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

Obviously, if ($\beta 3$) happens then $\varphi_{k_2}(k_2)$ is indeed defined. Suppose, ($\beta 4$) happens. We have to show that $\varphi_{k_2}(k_2)\uparrow$. 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 $L(G_{j_{\ell+y}}) = L_{4\langle j_r,k_2,0\rangle}$.

Then M fails to infer $L_{4\langle j_r,k_2,\Phi_{j_r}(j_r)+\Phi_{k_2}(k_2)\rangle+2}$ 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 $L_{4\langle j_r,k_2,0\rangle}$ and of a text $\tilde{t}$ for $L_{4\langle j_r,k_2,\Phi_{j_r}(j_r)+\Phi_{k_2}(k_2)\rangle+2}$. On the other hand, when successively fed $\tilde{t}$, the IIM M sometimes outputs $j_{\ell+y}$, and $L(G_{j_{\ell+y}}) = L_{4\langle j_r,k_2,0\rangle}$. Since $L_{4\langle j_r,k_2,0\rangle} \not\subseteq L_{4\langle j_r,k_2,\Phi_{j_r}(j_r)+\Phi_{k_2}(k_2)\rangle+2}$, we directly see that M violates the strong-monotonicity constraint.

Case 2. The hypothesis $j_{\ell+y}$ does not satisfy $L(G_{j_{\ell+y}}) = L_{4\langle j_r,k_2,0\rangle}$.

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 $L_{4\langle j_r,k_2,\ell\rangle+1}$ 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 $L_{4\langle j_r,k_2,\ell\rangle+1}$, too. On the other hand, $a^f b^{\ell+1} \notin L_{4\langle j_r,k_2,\ell\rangle+1}$. Consequently, $L(G_j) \not\subseteq L_{4\langle j_r,k_2,\ell\rangle+1}$, a contradiction.

Finally, since $j \neq j_{\ell+y}$, M has already performed at least one mind change when successively fed $\hat{t}_{\ell+y}$. Hence, $\hat{t}_{\ell+y}$ is an initial segment of a text for $L_{4\langle j_r,k_2,0\rangle}$ as well as for $L_{4\langle j_r,k_2,\ell+\Phi_{k_2}(k_2)\rangle+3}$. In accordance with $\mathcal{L}$'s definition we additionally have $L_{4\langle j_r,k_2,0\rangle} \subset L_{4\langle j_r,k_2,\ell+\Phi_{k_2}(k_2)\rangle+3}$. Thus, we may extend $\hat{t}_{\ell+y}$ with $a^f b^{\ell+y+1}$, $a^f b^{\ell+y+2}$, ... until M learns $L_{4\langle j_r,k_2,0\rangle}$. This forces M to change its mind again. Afterwards, we present $a^f e^{\ell+\Phi_{k_2}(k_2)}$, and hence, one more mind change has to occur. Thus, $\mathcal{L} \notin CSMON_2{-}TXT$. This contradiction proves Claim 3. Thus, Lemma 4 is shown, and the theorem follows. q.e.d.

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

Corollary 30. $EMON_1{-}TXT \setminus SMON{-}TXT \neq \emptyset$.

Proof. The indexed family $\mathcal{L}$ defined above belongs to $ELIM_1{-}TXT$. Hence, we

have to argue that $\mathcal{L} \notin SMON-TXT$. This is a direct consequence of Lemma 3 in the above proof. q.e.d.

In the next subsection we study monotonic inference.

7.2. 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 31. *$\lambda LIM_1-TXT = \lambda MON_1-TXT$ for all $\lambda \in \{E, \varepsilon, C\}$,*

Proof. Let $\mathcal{L}$ be any indexed family such that $\mathcal{L} \in \lambda LIM_1-TXT$, where $\lambda \in \{E, \varepsilon, C\}$. Hence, there are a hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ and an IIM M that λLIM_1-TXT–infers $\mathcal{L}$ with respect to $\mathcal{G}$. Consequently, when fed any text of any language $L \in \mathcal{L}$ the IIM M performs at most one mind change. Suppose, first M outputs k, and then it changes its mind to j. Hence, j has to be a correct guess for L, i.e., we have $L = L(G_j)$. Therefore, we directly obtain $L(G_k) \cap L \subseteq L(G_j) \cap L = L$. Hence, M monotonically infers $\mathcal{L}$. q.e.d.

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 32. *For every $n \in \mathbb{N}$, $n \geq 2$, there exists an indexed family such that*

(1) $\mathcal{L} \in MON_{n+1}-TXT \setminus MON_n-TXT$,

(2) $\mathcal{L} \in EWMON_n-TXT$.

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 (1993b). The desired indexed family is defined as follows. For all $k \in \mathbb{N}$, we set $L_{4k} = \{a^k b^z \mid z \in \mathbb{N}^+\}$ and $L_{4k+1} = L_{4k} \cup \{b^k a\}$. In order to define the remaining languages we distinguish the following cases:

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

Then we set $L_{4k+2} = L_{4k+3} = L_{4k+1}$.

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

Then, let $m = \Phi_k(k)$, and let $\hat{L}_k = \{a^k b^z \mid 1 \leq z \leq m\} \cup \{b^k a\}$. We set $L_{4k+2} = \hat{L}_k \cup \{a^k c^m\}$, and $L_{4k+3} = \hat{L}_k \cup \{a^k c^m,\ a^k d^m\}$.

Obviously, $\mathcal{L} = (L_z)_{z\in\mathbb{N}}$ is an indexed family of recursive languages. To see this take into consideration that, for instance, $a^k b^z \in L_{4k+2}$ iff $\Phi_k(k) \geq z$. But $\Phi_k(k) \geq z$ is uniformly decidable for any $z,\ k \in \mathbb{N}$.

Claim A. $\mathcal{L} \in EMON_3-TXT$.

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

$M(t_x) =$ "Determine the unique k such that $a^k b^z \in t_x^+$ for some $m \in \mathbb{N}$. If $b^k a \notin t_x^+$ output $4k$. Otherwise, goto (A).

(A) If $a^k d^m \in t_x^+$ and $t_x^+ \subseteq L_{4k+3}$, then output $4k+3$.
If $a^k c^m \in t_x^+$ and $t_x^+ \subseteq L_{4k+2}$, then output $4k+2$.
Otherwise, output $4k+1$."

Obviously, M monotonically infers $\mathcal{L}$. In the worst case, M changes its mind three times, namely it outputs successively the hypotheses $4k$, $4k+1$, $4k+2$, and $4k+3$. It is easy to verify that each of these mind changes satisfies the monotonicity requirement.

Claim B. $\mathcal{L} \in EWMON_2{-}TXT$.

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

$M(t_x) =$ "Determine the unique k such that $a^k b^z \in t_x^+$ for some $z \in \mathbb{N}^+$. If $b^k a \notin t_x^+$ output $4k$. Otherwise, goto (A).

(A) If $t_x^+ \subseteq L_{4k+2}$ output $4k+2$.
If $a^k d^m \in t_x^+$ output $4k+3$.
Otherwise, i.e., a string $a^k b^z \notin L_{4k+2}$ occurred, output $4k+1$."

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

Note that the IIM M defined in the latter proof may subsequently output the hypotheses $4k$, $4k+2$, and $4k+1$ when fed a text for L_{4k+1}. It is easy to verify that the latter mind change violates the monotonicity requirement. Moreover, it is easy to argue that $\mathcal{L} \notin EMON_2{-}TXT$. But again, we show a slightly stronger result.

Claim C. $\mathcal{L} \notin MON_2{-}TXT$.

Claim C follows by contraposition of Lemma 1.

***Lemma* 1**. *Given any class preserving hypothesis space $\mathcal{G}$ and any program for an IIM M witnessing $\mathcal{L} \in MON_2{-}TXT$ with respect to $\mathcal{G}$, one can effectively define a total-recursive predicate ψ solving the halting problem.*

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

$\psi(k) =$ "Let t be the lexicographically ordered text for L_{4k}. For $x = 0,\ 1,\ ...$, compute $M(t_x)$ until the first index z is found such that $j = M(t_z)$ satisfies $a^k b \in L(G_j)$, and $b^k a \notin L(G_j)$.

(A) For $r = 1,\ 2,\ \ldots$ simulate M, when fed the text $\hat{t} = t_z,\ b^k a,\ a^k b,\ ...,\ a^k b^r$ for L_{4k+1}, until the first index y is found such that $j_{z+1+y} = M(\hat{t}_{z+1+y})$ satisfies $b^k a \in L(G_{j_{z+1+y}})$.

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

Since M has to infer the languages L_{4k} as well as L_{4k+1}, it is easy to verify that the procedure defined above terminates for every $k \in \mathbb{N}$. Hence, ψ is total-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}. Due to the definition of $\mathcal{L}$, $b^k a \notin L(G_j)$ together with $a^k b \in$

$L(G_j)$ implies $L(G_j) = L_{4k}$. Since M monotonically infers L_{4k+1} from $\hat{t}$ and $b^k a \in L(G_{j_{z+1+y}})$, we obtain $L(G_{j_{z+1+y}}) = L_{4k+1}$. Otherwise, M violates the monotonicity constraint when inferring L_{4k+1} on its text $\hat{t}$. 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_{4k+2} because $\Phi_k(k) = m > z + 1 + y$. Finally, since $L_{4k+2} \subset L_{4k+3}$, it is easy to verify that $\hat{t}_{z+1+y}$ can be extended to a text for L_{4k+3} such that M has to perform at least two additional mind changes in order to infer L_{4k+3} from this text. This contradicts our assumption that M monotonically infers $\mathcal{L}$ with at most two mind changes. Therefore, $\varphi_k(k)$ is undefined, if $\psi(k) = 0$. Hence, the predicate ψ solves the halting problem for the φ-system. q.e.d.

Refining *mutatis mutandis* the latter proof analogously as the demonstration of Theorem 27 has been extended to show Theorem 29, one obtains the following result.

Theorem 33. *For every $n \geq 2$ there exists an indexed family such that*

(1) $\mathcal{L} \in MON_{n+1}{-}TXT \setminus MON_n{-}TXT$,

(2) $\mathcal{L} \in EWMON_2{-}TXT$.

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

7.3. Weak-Monotonic Learning

Finally, we consider weak-monotonic learning. Possible relaxations include dual weak-monotonic learning as well as learning in the limit. However, much less is known. First, Theorem 31 directly implies $LIM_1{-}TXT = WMON_1{-}TXT$ as well as $ELIM_1{-}TXT = EWMON_1{-}TXT$, since $\lambda MON{-}TXT \subset \lambda WMON{-}TXT$ for $\lambda \in \{E, \varepsilon\}$. On the other hand, it is even open whether or not $CMON_1{-}TXT \subseteq CWMON_1{-}TXT$. Hence, showing $CLIM_1{-}TXT = CWMON_1{-}TXT$ requires a separate proof that is still missing. Nevertheless, we succeeded to obtain results that shed considerable light on the power of learning with at most one mind change.

Theorem 34.

(1) $MON_1{-}TXT \setminus EWMON{-}TXT \neq \emptyset$

(2) $ELIM_2{-}TXT \setminus WMON{-}TXT \neq \emptyset$

(3) $CMON_1{-}TXT \setminus WMON{-}TXT \neq \emptyset$

Proof. Lange and Zeugmann (1993b) proved $LIM_1{-}TXT \setminus EWMON{-}TXT \neq \emptyset$, and recently Lange (1994) shows $CLIM_1{-}TXT \setminus WMON{-}TXT \neq \emptyset$. Combining these results with Theorem 31 we directly get Assertion (1) and (3). Finally, for a proof of Assertion (2) we refer the reader to Lange (1994). q.e.d.

Consequently, relaxing the weak-monotonicity constraint may considerably increase the inference capabilities.

We conclude this section with further problems that remain open. First of all, it would be very interesting to answer the following question. Does there exist a $k \in \mathbb{N}$

such that $CLIM_k \setminus CWMON \neq \emptyset$? Of course, one should ask similar questions for dual monotonic and especially for dual weak-monotonic learning.

Furthermore, our results show that a relaxation of the corresponding monotonicity demands sometimes yields a significant speed-up of the learning process. Hence, it seems highly desirable to investigate necessary and sufficient conditions $\mathcal{C}_{csmon}$, $\mathcal{C}_{mon}$, and $\mathcal{C}_{wmon}$ 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 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, 1994)). We will see what the future brings concerning these problems.

8. Degrees of Order Independence

In this section we study the question whether or not the *order* of information presentation does really influence the capabilities of IIMs. Since an IIM is required to learn the target language from every text (informant) for it, one may conjecture that an IIM mainly extracts the range of the information fed to it, thereby neglecting the length and order of the data sequence it reads. While this is true for learning from informant, the situation considerably changes for inference from positive data. A first explanation for this phenomenon can be derived from the fact that, when fed an informant, an IIM can *decide* whether or not is has already seen a *complete* initial segment. Then it ignores all the other data fed to it, and behaves like an IIM learning from the *lexicographically* ordered informant. Clearly, when exclusively learning from positive data, an IIM never knows whether it possesses a complete initial segment. But this is only part of the story as we shall see. Next we define two modes of order independence.

Definition 11. (Wexler and Culicover, 1980, Sec. 2.2) *Let $\mathcal{L}$ be an indexed family.* ***An IIM is said to be set-driven*** *with respect to $\mathcal{L}$ iff its output depends only on the range of its input; that is, iff $M(t_x) = M(\hat{t}_y)$ for all $x, y \in \mathbb{N}$, all texts t, $\hat{t} \in \bigcup_{L \in range(\mathcal{L})} text(L)$ provided $t_x^+ = \hat{t}_y^+$.*

Schäfer-Richter (1984) as well as Fulk (1990), later, and independently proved that set-driven IIMs are less powerful than unrestricted ones. Fulk (1990) interpreted the weakening in the learning power of set-driven IIMs by the need of IIMs for time to "reflect" on the input. However, this time cannot be bounded by any *a priori* fixed computable function depending exclusively on the size of the range of the input, since otherwise set-drivenness would not restrict the learning power. Indeed, Osherson, Stob and Weinstein (1986) proved that any *non-recursive* IIM M may be replaced by a *non-recursive* set-driven IIM $\hat{M}$ learning at least as much as M does. On the other hand, the weakness of set-driven inference has been proved in a domain that allows self-referential arguments. Since this proof technique does not apply in the setting of indexed families, the problem whether or not set-drivenness constitutes a severe restriction in this domain remained open. But before starting our survey of results

we consider a natural weakening of Definition 11.

Definition 12. (Schäfer-Richter, 1984; Osherson et al., 1986)) *Let $\mathcal{L}$ be an indexed family.* ***An IIM is said to be rearrangement-independent*** *iff its output depends only on the range and on the length of its input; that is, iff $M(t_x) = M(\hat{t}_x)$ for all $x \in \mathbb{N}$, all texts t, $\hat{t} \in \bigcup_{L \in range(\mathcal{L})} text(L)$ provided $t_x^+ = \hat{t}_x^+$.*

We make the following convention. For all the learning models in this paper we use the prefix $s-$, and $r-$ to denote the learning model restricted to set-driven and rearrangement-independent IIMs, respectively. For example, $s-LIM$ denotes the collection of all indexed families that are LIM–inferable by some set-driven IIM.

Fulk (1990) proved that rearrangement-independence can be always achieved when learning in the limit is concerned. However, Fulk's proof technique does not preserve any of the monotonicity constraints defined in Section 2. On the other hand, the first result concerning order-independence with respect to the inferability of indexed families goes back to Angluin (1980a). As already mentioned in Section 3 she characterized learning in the limit using families of *recursively enumerable* finite and non-empty tell-tales. In particular, the IIM defined in the sufficiency part of her characterization theorem establishes that rearrangement-independence does not constitute a restriction for $ELIM$ (cf. Section 3, Example 5). In Lange and Zeugmann (1993b) we proved that $ELIM = LIM = CLIM$. Hence, learning in the limit can be always achieved by rearrangement-independent IIMs. Inspired by Angluin's (1980b) work we characterized conservative, monotonic, strong-monotonic, and finite learning in terms of *recursive* finite and non-empty tell-tales (cf. Lange and Zeugmann (1992)). These results directly yield that $r-EFIN = FIN$, and $r-SMON = SMON$. However, all remaining questions required special attention. In the following subsection we survey results concerning set-drivenness.

8.1. Learning with Set-driven IIMs

We start with finite learning. The next theorem in particular states that finite learning is invariant with respect to the specific choice of the hypothesis space. Moreover, for every hypothesis space comprising the target indexed family $\mathcal{L}$ there is a *set-driven* IIM that finitely learns $\mathcal{L}$.

Theorem 35. $EFIN = FIN = CFIN = s-EFIN$

Proof. $EFIN = FIN = CFIN$ is due to Lange and Zeugmann (1993c).

The main ingredient for the proof of $EFIN = s-EFIN$ is the following characterization of finite learning (cf. Lange and Zeugmann (1992)).

Theorem 36. *Let $\mathcal{L}$ be an indexed family. Then: $\mathcal{L} \in FIN-TXT$ if and only if there is a recursively generable family $(T_j)_{j \in \mathbb{N}}$ of finite non-empty sets such that*

(1) *for all $j \in \mathbb{N}$, $T_j \subseteq L_j$,*

(2) *for all k, $j \in \mathbb{N}$, if $T_k \subseteq L_j$, then $L_j = L_k$.*

Using this recursively generable family $(T_j)_{j \in \mathbb{N}}$ we define a IIM M witnessing $\mathcal{L} \in s-EFIN-TXT$. Let $L \in \mathcal{L}$, $t \in text(L)$, and $x \in \mathbb{N}$.

$M(t_x) =$ " If $x = 0$ or $x > 0$ and M when, fed successively t_{x-1}, does not stop, then execute stage x.

Stage x: Search for the least j such that $t_x^+ \subseteq L_j$. Test whether or not $T_j \subseteq t_x^+$.
In case it is, output j and stop.
Otherwise, request the next input and output nothing."

We sketch only the idea behind the proof. The IIM defined above searches for the first language that comprises t_x^+. Since t is a text of some language from $\mathcal{L}$, this *unbounded* search terminates. Then M tests whether or not the relevant tell-tale belongs to the range of the initial text segment it has been fed. Hence, intuitively it is clear that M is set-driven. It remains to show that M has to stop sometimes. This part of the proof mainly exploits Property (2) of the tell-tale family. M might fail to stop provided the first index j that is found by M is proper superset of the target language L and $T_j \not\subseteq L$. But this is impossible, since then $T_L \subseteq L_j$, where T_L is any tell-tale for the target language L. By Property (2) we conclude $L = L_j$, a contradiction.

For a formal proof of M $s{-}EFIN{-}TXT$–learns $\mathcal{L}$ the reader is referred to Lange and Zeugmann (1993e). q.e.d.

The proof sketch given above makes its clear why set-drivenness may constitute a severe restriction. The main problem is just a suitable restriction of the actual search space an IIM may use to compute its actual guess. If the topological structure of $\mathcal{L}$ is more complicated than in case of finite learning, then the method presented above fails. The next theorem shows that any other method fails as well.

Theorem 37. $s{-}CLIM{-}TXT \subset ELIM{-}TXT = LIM{-}TXT = CLIM{-}TXT$

Proof. The part $ELIM{-}TXT = LIM{-}TXT = CLIM{-}TXT$ is due to Lange and Zeugmann (1993c). Again, the part $s{-}CLIM{-}TXT \subset ELIM{-}TXT$ is only sketched.

As a matter of fact, the proof technique introduced in Section 5 is powerful enough to show the desired result. The desired separation can be obtained by using the indexed family $\mathcal{L}$ of Example 6. Thus, $\mathcal{L}$ is defined as follows. For all $k \in \mathbb{N}$ we set $L_{\langle k,0\rangle} = \{a^k b^n \mid n \in \mathbb{N}^+\}$. For all $k \in \mathbb{N}$ and all $j \in \mathbb{N}^+$ we distinguish the following cases:

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

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

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

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

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

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

Any set-driven IIM that might learn $\mathcal{L}$ has to overcome two difficulties. First, it has to find a hypothesis, and second, it has to detect whether or not its actual guess is *overgeneralized*. Intuitively, detecting overgeneralization forces M to handle the halting problem. Hence, it may try to avoid overgeneralized hypothesises. But again, this forces M to search the *least* language with respect to set inclusion comprising t_x^+.

Since the halting problem is algorithmically unsolvable, M cannot decide whether or not to continue its search for a number of a least language. But if it gives up, it might overgeneralize, and we are back to Case 1. This can be well formalized, and indeed one can show that any set-driven IIM learning $\mathcal{L}$ directly yields an algorithm solving the halting problem (cf. Lange and Zeugmann (1993e)).

On the other hand, it is not hard to prove that $\mathcal{L}$ can be inferred in the limit with respect to the hypothesis space $\mathcal{L}$. The main idea can be described as follows. The desired IIM M uses the *length* of its actual input to test whether $\Phi_k(k)$ might be defined. As long as $\Phi_k(k)$ does not turn out to be defined, M simply outputs the corresponding index $\langle k, 0\rangle$, where k can be easily computed from t_0. In case $\Phi_k(k)$ happens to be defined, M can *effectively* search for the least language in $\mathcal{L}$ that comprises t_x. q.e.d.

As the latter theorem shows, sometimes there is no way to design a set-driven IIM. However, with the following theorems we mainly intend to show that the careful choice of the hypothesis space deserves special attention whenever set-drivenness is desired.

Theorem 38. *There is an indexed family $\mathcal{L}$ such that*

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

(2) *no set-driven IIM M $LIM\!-\!TXT$–infers $\mathcal{L}$,*

(3) *there are a hypothesis space $\mathcal{G}$ and an IIM M witnessing $\mathcal{L} \in s$–$CSMON$–TXT with respect to $\mathcal{G}$.*

As we have seen, set-drivenness constitutes a severe restriction. While this is true in general as long as exact and class preserving learning is considered, the situation looks differently in the class comprising case. On the one hand, learning in the limit cannot always be achieved by set-driven IIMs (cf. Theorem 37). On the other hand, conservative learners may always be designed to be set-driven, if the hypothesis space is appropriately chosen.

Theorem 39. *s–$CCONSERVATIVE$–$TXT = CCONSERVATIVE$–TXT*

Again, we only sketch the main ideas of the proof, and refer the interested reader to Lange and Zeugmann (1993e) for any detail. The proof is partitioned into two parts. The first part establishes the equality of class comprising conservative and class comprising, rearrangement-independent conservative learning. The main ingredients into this proof are the characterization of *CCONSERVATIVE–TXT* (cf. Section 3, Theorem 5) as well as a technically simple, but powerful modification of the corresponding tell-tale family (cf. Section 4, Theorem 9). For the sake of readability, we recall these results.

Let $\mathcal{L} \in CCONSERVATIVE\!-\!TXT$. Then there exist a space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ of hypotheses and a recursively generable tell-tale family $(T_j)_{j\in\mathbb{N}}$ of finite and non-empty sets such that

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

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

(3) for all $j, k \in \mathbb{N}$, if $T_j \subseteq L(G_k)$, then $L(G_k) \not\subset L(G_j)$.

Using this tell-tale family, we define a new recursively generable family $(\hat{T}_j)_{j\in\mathbb{N}}$ of finite and non-empty sets that allows the design of a rearrangement-independent IIM inferring $\mathcal{L}$ conservatively with respect to $\mathcal{G}$. But surprisingly enough, we can even do better, namely, we can define an IIM witnessing $\mathcal{L}(\mathcal{G}) \in r\text{–}ECONSERVATIVE-TXT$. For all $j \in \mathbb{N}$ we set $\hat{T}_j = \bigcup_{n\leq j} T_n \cap L(G_j)$. Note that the new tell-tale family fulfills Properties (1) through (3) above.

Now, the wanted IIM can be defined as follows: Let $L \in \mathcal{L}(\mathcal{G})$, $t \in \mathit{text}(L)$, and $x \in N$.

$M(t_x) =$ "Generate $\hat{T}_k$ for all $k \leq x$ and test whether $\hat{T}_k \subseteq t_x^+ \subseteq L(G_k)$. In case there is one k fulfilling the test, output the minimal one, and request the next input. Otherwise, output nothing and request the next input."

Obviously, M is rearrangement-independent. We omit the proof that M *ECONSERVATIVE–TXT*–learns $\mathcal{L}(\mathcal{G})$.

The second part of the proof establishes set-drivenness. For that purpose, we define a new hypothesis space $\tilde{\mathcal{G}} = (\tilde{G}_j)_{j\in\mathbb{N}}$ as well as a new IIM $\tilde{M}$. The basis for these definitions are the hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$, and the IIM M described above. The hypothesis space $\tilde{\mathcal{G}}$ is the canonical enumeration of all grammars from $\mathcal{G}$ and all finite languages over the underlying alphabet Σ. Before defining the IIM $\tilde{M}$, we introduce the notion of *repetition free* text $rf(t)$. Let $t = s_0, s_1, ...$ be any text. We set $rf(t_0) = s_0$ and proceed inductively as follows: For all $x \geq 1$, $rf(t_{x+1}) = rf(t_x)$, if $s_{x+1} \in rf(t_x)^+$, and $rf(t_{x+1}) = rf(t_x), s_{x+1}$ otherwise. Obviously, given any initial segment t_x of a text t one can effectively compute $rf(t_x)$. Now we are ready to present the definition of $\tilde{M}$. Let $L \in \mathcal{L}(\mathcal{G})$, $t \in \mathit{text}(L)$, and $x \in \mathbb{N}$.

$\tilde{M}(t_x) =$ "Compute $rf(t_x)$. If M on input $rf(t_x)$ outputs a hypothesis, say j, then output the canonical index of j in $\tilde{\mathcal{G}}$ and request the next input.
Otherwise, output the canonical index of t_x^+ in $\tilde{\mathcal{G}}$ and request the next input."

Intuitively, it is clear that $\tilde{M}$ is set-driven. We omit the proof that $\tilde{M}$ *CCONSERVATIVE–TXT*–learns $\mathcal{L}$ with respect to $\tilde{\mathcal{G}}$. q.e.d.

The latter theorem allows a nice corollary that we present next. In particular, this corollary shows that the IIM $\tilde{M}$ defined above can be transformed into an IIM M that learns much more than one might expect.

Corollary 40. *Let $\mathcal{L} \in CCONSERVATIVE-TXT$. Then, there exists a hypothesis space $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ comprising $\mathcal{L}$ such that $\mathcal{L}(\hat{\mathcal{G}}) \in s\text{–}ECONSERVATIVE-TXT$.*

Proof. Let $\mathcal{L} \in CCONSERVATIVE-TXT$. Furthermore, by the latter theorem, there are an IIM $\tilde{M}$ and a hypothesis space $\tilde{\mathcal{G}}$ such that $\tilde{M}$ $s-CCONSERVATIVE$–TXT–infers $\mathcal{L}$ with respect to $\tilde{\mathcal{G}}$.

Recall that $\tilde{\mathcal{G}}$ is a canonical enumeration of $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ satisfying $\mathcal{L} \subseteq \mathcal{L}(\mathcal{G})$ and of all finite languages over the underlying alphabet. Without loss of generality we may assume that $\tilde{\mathcal{G}}$ fulfills the following property. If j is even, then $L(\tilde{G}_j) \in \mathcal{L}(\mathcal{G})$. Hence, $\tilde{M}$ $s-CCONSERVATIVE-TXT$–learns $L(\tilde{G}_j)$ with respect to $\tilde{\mathcal{G}}$. Otherwise, $L(\tilde{G}_j)$ is a finite language.

We start with the definition of the desired hypothesis space $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ If j is even, then we set $\hat{G}_j = \tilde{G}_j$. Otherwise, we distinguish the following cases. If M when fed the lexicographically ordered enumeration of all strings in $L(\tilde{G}_j)$ outputs the hypothesis j, then we set $\hat{G}_j = \tilde{G}_j$. In case it does not, we set $\hat{G}_j = \tilde{G}_{j-1}$.

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

$M(t_x) =$ "Simulate $\tilde{M}$ on input t_x. If $\tilde{M}$ does not output any hypothesis, then output nothing and request the next input.
Otherwise, let $\tilde{M}(t_x) = j$. Output j and request the next input."

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

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

8.2. Learning with Rearrangement-Independent IIMs

In this section we study the impact of rearrangement-independence on the learning power of IIMs. Recall that $r-ELIM-TXT = CLIM-TXT$ as well as $r-SMON-TXT = SMON-TXT$. So what about $ESMON-TXT$? By answering this question another proof technique comes into the play. The key idea consists in applying Theorem 17. Hence, we may use a rearrangement-independent IIM $\hat{M}$ as well as a class preserving hypothesis space $\mathcal{G}$ such that $\mathcal{L} \in r{-}SMON$ with respect to $\mathcal{G}$ is witnessed by M. Due to that Theorem there exists a strong-monotonic limiting recursive compiler f from $\mathcal{G}$ into $\mathcal{L}$. Therefore, all we have to do is to combine the IIM $\hat{M}$ and the strong-monotonic limiting recursive compiler f. And indeed, this idea goes through. Thus, $ESMON{-}TXT = r{-}ESMON{-}TXT$.

Moreover, the latter result cannot be improved as the next theorem states. Furthermore, the sketched proof technique does not apply to monotonic language learning, and so does any other proof technique. As a matter of fact, monotonic inference is *very* sensitive with respect to the order in which the input data are presented.

Theorem 41.

(1) $s{-}EMON{-}TXT \subset r{-}EMON{-}TXT \subset EMON{-}TXT$,

(2) $s{-}MON{-}TXT \subset r{-}MON{-}TXT \subset MON{-}TXT$.

Finally, we consider rearrangement-independence in the context of exact and class preserving conservative learning. Since conservative learning is exactly as powerful as weak-monotonic one, by the latter theorem one might expect that rearrangement-independence is a severe restriction under the weak-monotonic constraint, too. On

the other hand, looking at Theorem 39 we see that conservative learning has its peculiarities. And indeed, exact and class preserving learning can always be performed by rearrangement-independent IIMs. In order to prove this, we first characterize *ECONSERVATIVE* in terms of finite tell-tales. We present this theorem separately, since it is interesting in its own right.

Theorem 42. *Let $\mathcal{L}$ be an indexed family. Then, $\mathcal{L} \in ECONSERVATIVE-TXT$ if and only if there exists a recursively generable family $(T_j^y)_{j,y\in\mathbb{N}}$ of finite sets such that*

(1) *for all $L \in \mathcal{L}$ there exists a j with $L_j = L$ and $T_j^y \neq \emptyset$ for almost all $y \in \mathbb{N}$,*

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

(3) *for all $j,\ y,\ z \in \mathbb{N}$, $\emptyset \neq T_j^y \subseteq L_z$ implies $L_z \not\subset L_j$.*

For a proof, the reader is referred to Lange and Zeugmann (1993e).

Finally, applying the same technique as described in the proof of Theorem 39 one may modify *mutatis mutandis* the tell-tale family $(T_j^y)_{j,y\in\mathbb{N}}$ appropriately. Then, the new family as well as a suitable modification of the IIM defined in the proof of Theorem 39 directly yield the rearrangement-independence of exact conservative learning. Moreover, the same ideas are powerful enought to show the analogous result for class preserving conservative inference (cf. Section 4, Theorem 9). Hence, we have the following theorem.

Theorem 43.

(1) $r-ECONSERVATIVE-TXT = ECONSERVATIVE-TXT$,

(2) $r-CONSERVATIVE-TXT = CONSERVATIVE-TXT$.

With the following figure we summarize the results surveyed in this section and point to questions that remain open. We shall discuss them and some less obvious ones in Section 9.

For every model of learning LT mentioned "*rearrangement-independence* $+$" indicates $r-LT = LT$ as well as $s-LT \subset LT$. "*Rearrangement-independence* $-$" implies $s-LT \subset r-LT \subset LT$ whereas "*set-drivenness* $+$" should be interpreted as $s-LT = LT$ and, therefore, $r-LT = LT$, too.

	exact learning	class preserving learning	class comprising learning
FIN	*set drivenness* +	*set drivenness* +	*set drivenness* +
$SMON$	*rearrangement independence* +	*rearrangement independence* +	**?**
MON	*rearrangement independence* −	*rearrangement independence* −	**?**
$WMON$	*rearrangement independence* +	*rearrangement independence* +	*set drivenness* +
LIM	*rearrangement independence* +	*rearrangement independence* +	*rearrangement independence* +

9. Outlook

We started our guided tour with several question that are closely related to the design of "natural" learning algorithms. Therefore, we continue with a short discussion of our results in this regard. Mathematically sound formalizations of learning by generalization and specialization have been introduced. Furthermore, the models of weak-monotonic learning and of dual weak-monotonic inference formalized the problem to what extend non-monotonic reasoning has to be incorporated into the learning process. As we have seen, superior learning algorithms can be designed if and only if most of the monotonicity demands are dropped (cf. e.g. Section 5, Theorem 11 as well as Figure 1 and 2). Consequently, our results provide considerable evidence that learning has to be performed, at least to some extend, incorporating non-monotonic reasoning.

Furthermore, the characterizations obtained provide a unifying framework to all types of monotonic learning. Hence, we achieved considerable insight into the problem what is more appropriate, learning by specialization or learning by generalization. All the differences between these two global learning strategies can be expressed in terms of properties the hypothesis space and the corresponding finite tell-tale sets must satisfy. These results strongly recommend to study particular properties of indexed families that can be expressed by suitable descriptions of the objects to be learned and by finite tell-tale sets (cf. e.g. Example 4). As we have seen, it is mainly the interplay between the properties of the relevant hypothesis space and the relevant tell-tale sets that makes or does not make a learning problem solvable. The latter assertion even remains true, if additional postulates of naturalness are involved. Our theorem stating $s - CCONSERVATIVE - TXT = CCONSERVATIVE\text{–}TXT$ may serve as an illustrative example (cf. Section 8, Theorem 39).

However, some intriguing questions concerning order independence remain open. Two of them are presented in the figure above. Additionally, it would be highly desirable to elaborate characteristic conditions under what circumstances set-drivenness does not restrict the learning power. We expect that such characterizations might

allow much more insight into the problem how to handle simultaneously both, finite and infinite languages in the learning process [2]. Next, as we have seen, an algorithmically solvable learning problem might become infeasible, if one tries to solve it with set-driven IIMs. On the other hand, when dealing with particular learning problems it might often be possible to design a set-driven learning algorithm solving it. But what about the complexity of learning in such circumstances? More precisely, we are interested in knowing whether the "high-level" theorem separating set-driven learning from unrestricted one, has an analogue in terms of complexity theory. For example, it is well conceivable that an indexed family $\mathcal{L}$ may be learned in polynomial time but no set-driven algorithm can efficiently infer $\mathcal{L}$ provided $\mathcal{P} \neq \mathcal{NP}$.

Moreover, our results suggest some further avenues of research. All learning models described in this paper dealt with *passive* inference, i.e., the IIM itself has no influence to the data it is fed. Hence, it seems to be very promising to study *active* learning, too (cf. Angluin (1992) and the references therein). The most common types of queries are *equivalence* and *membership* queries. Clearly, each indexed family can be learned by equivalence queries alone. However, this approach may lead to non-efficient solutions. On the other hand, there are some results showing that particular indexed families are learnable in polynomial time using membership and equivalence queries (cf. e.g. Ishizaka (1989)). Obviously, the crucial point is to determine what membership queries the learner should ask. We conjecture that tell-tales might be very helpful to solve the latter problem.

Finally, it seems very promising to study the learnability of indexed families within probabilistic models of inductive inference. In the setting of inductive inference of recursive functions Freivalds, Kinber and Wiehagen (1988) proved the following interesting result. There are hypothesis spaces $\mathcal{H}$ such that non-exactly learnable function classes might become inferable with probability 1 with respect to $\mathcal{H}$. It would be interesting to know whether or not similar effects might occur in the setting of learning recursive languages. Furthermore, Wiehagen, Freivalds and Kinber (1984) proved the superiority of probabilistic inference algorithms with respect to the number of allowed mind changes. Again, we are interested in learning whether or not these results extend to our setting.

10. References

Adleman, L.M., and Blum, M. (1991), Inductive inference and unsolvability, *Journal of Symbolic Logic* **56**, 891 – 900.

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

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

Angluin, D. (1992), Computational learning theory: Survey and selected bibliography, *in* "Proceedings 24th Annual ACM Workshop on Theory of Computing," pp. 351 – 369, ACM Press.

[2] For recent results see Lange and Zeugmann (1996).

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

Angluin, D., and Smith, C.H. (1987), Formal inductive inference, *in* "Encyclopedia of Artificial Intelligence" (St.C. Shapiro, Ed.), Vol. 1, pp. 409 – 418, Wiley-Interscience Publication, New York.

Anthony, M. and Biggs, N. (1992), "Computational Learning Theory," Cambridge University Press, Cambridge.

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

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

Arikawa, S., Kuhara, S., Miyano, S., Mukouchi, Y., Shinohara, A. and Shinohara, T. (1992), A machine discovery from amino acid sequences by decision trees over regular patterns, *in* Proceedings International Conference on Fifth Generation Computer Systems, Vol. 2, pp. 618 – 625, Institute for New Generation Computer Technology (ICOT), Tokyo, Japan.

Barzdin, Ya.M. (1974), Inductive inference of automata, functions and programs, *in* "Proceedings International Congress of Math.," Vancouver, pp. 455 – 460.

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

Barzdin, Ya.M., and Freivalds, R.V. (1974), Прогнозирование и предельный синтез эффективно перечислимых классов функций, *in* "Теория Алгоритмов и Программ," Vol. 1 (Ya. M. Barzdin, ed.) Latvian State University, Riga, pp. 101 – 111.

Barzdin, Ya.M., Kinber, E.B., and Podnieks, K.M. (1974), Об ускорении синтеза и прогнозирования функций, *in* "Теория Алгоритмов и Программ," Vol. 1 (Ya.M. Barzdin, Ed.) Latvian State University, Riga, pp. 117 – 128.

Berwick, R. (1985), "The Acquisition of Syntatic Knowledge," MIT Press, Cambridge, Massachusetts.

Blum, A., and Singh, M. (1990), Learning functions of k terms, *in* "Proceedings 3rd Workshop on Computational Learning Theory, July 1990, Rochester," (M. Fulk and J. Case, Eds.), pp. 144 – 153, Morgan Kaufmann Publishers Inc., San Mateo.

Brewka, G. (1991), "Nonmonotonic Reasoning: Logical Foundations of Commonsense," Cambridge University Press, Cambridge.

Case, J. (1988), The power of vacillation, *in* "Proceedings 1st Workshop on Computational Learning Theory, August 1988, Boston," (D. Haussler and L. Pitt, Eds.), pp. 196 – 205, Morgan Kaufmann Publishers Inc., San Mateo.

Case, J., and Lynes, C. (1982), Machine inductive inference and language identification, *in* "Proceedings Automata, Languages and Programming, Ninth Colloquium, Aarhus, Denmark," (M. Nielsen and E.M. Schmidt, Eds.), Lecture Notes in Computer Science Vol. 140, pp. 107 – 115, Springer-Verlag, Berlin.

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

Freivalds, R., Kinber, E.B. and Wiehagen, R. (1988), Probabilistic versus deterministic inductive inference in nonstandard numberings, *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, **34** (1988), 531 – 539.

Freivalds, R., Kinber, E.B. and Wiehagen, R. (1992), Convergently versus divergently incorrect hypotheses in inductive inference, GOSLER–Report 02/92, January 1992, FB Mathematik und Informatik, TH Leipzig.

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

Fulk, M., and Case, J. (Eds.) (1990), Proceedings of the 3rd Annual Workshop on Computational Learning Theory, July 1990, Rochester, Morgan Kaufmann Publishers Inc., San Mateo.

Gasarch, W.I., and Velauthapillai, M. (1992), Asking questions versus verifiability, *in* "Proceedings 3rd International Workshop on Analogical and Inductive Inference," October 1992, Dagstuhl, (K.P. Jantke, ed.) Lecture Notes in Artificial Intelligence Vol. 642, pp. 197 – 213, Springer-Verlag, Berlin.

Gold, M.E. (1965), Limiting recursion, *Journal of Symbolic Logic*, **30** 28 – 48.

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

Haussler, D. (Ed.) (1992), Proceedings of the 5th Annual Workshop on Computational Learning Theory, July 1992, Pittsburgh, ACM Press, New York.

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

Ishizaka, H. (1989), Learning simple deterministic languages. *in* "Proceedings of the 2nd Annual Workshop on Computational Learning Theory, Santa Cruz, August 1989, (R. Rivest, D. Haussler and M.K. Warmuth, Eds.), pp. 162–174, Morgan Kaufmann Publishers Inc., San Mateo.

Jain, S., and Sharma, A. (1989), Recursion theoretic characterizations of language learning, The University of Rochester, Dept. of Computer Science, TR 281.

Jantke, K.P. (Ed.) (1989), "Proceedings 2nd International Workshop on Analogical and Inductive Inference, October 1989, Reinhardsbrunn Castle," Lecture Notes in Artificial Intelligence Vol. 397.

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

Jantke, K.P. (1991b), Monotonic and non-monotonic inductive inference of functions and patterns, *in* "Proceedings 1st International Workshop on Nonmonotonic and Inductive Logics, December 1990, Karlsruhe," (J. Dix , K.P. Jantke and P.H. Schmitt, Eds.), Lecture Notes in Artificial Intelligence Vol. 543, pp. 161 – 177, Springer-Verlag, Berlin.

Jantke, K.P. (Ed.) (1992), "Proceedings 3rd International Workshop on Analogical and Inductive Inference, October 1992, Dagstuhl Castle," Lecture Notes in Artificial Intelligence Vol. 642.

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

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

Kearns, M., and Pitt, L. (1989), A polynomial-time algorithm for learning k–variable pattern languages from examples, *in* "Proceedings 1st Annual Workshop on Computational Learning Theory, August 1988, Boston," (D. Haussler and L. Pitt, Eds.), pp. 196 –205, Morgan Kaufmann Publishers Inc., San Mateo.

Kinber, E.B. (1992), personal communication.

Kodratoff, Y., and Michalski, R.S. (1990), "Machine Learning, An Artificial Intelligence Approach," Vol. 3, Morgan Kaufmann Publishers Inc., San Mateo.

Lange, S. (1994), The representation of recursive languages and its impact on the efficiency of learning, *in* "Proceedings 7th Annual ACM Conference on Computational Learning Theory, New Brunswick, July 1994," (M. Warmuth, Ed.), pp. 256 – 267, ACM Press, New York.

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

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

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

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

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

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

Lange, S., and Zeugmann, T. (1993e), On the impact of order independence to the learnability of recursive languages, Research Report ISIS-RR-93-17E, Institute for Social Information Science, FUJITSU Laboratories Ltd, Numazu.

Lange, S., and Zeugmann, T. (1994), Characterization of language learning on informant under various monotonicity constraints, *Journal of Experimental and Theoretical Artificial Intelligence* **6**, 73 – 94.

Lange, S., and Zeugmann, T. (1995), Modeling incremental learning from positive data, Technical Report RIFIS-TR-CS-117, Research Institute of Fundamental Information Science (RIFIS), Kyushu University, Fukuoka.

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

Lange, S., and Zeugmann, T., and Kapur, S (1992), Class preserving monotonic language learning, GOSLER–Report 14/92, FB Mathematik und Informatik, TH Leipzig, appeared as:

Monotonic and dual monotonic language learning, *Theoretical Computer Science* **155** (1996), 365 – 410.

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

Michalski, R.S., Carbonell, J.G., and Mitchell, T.M. (1984), "Machine Learning, An Artificial Intelligence Approach," Vol. 1, Springer-Verlag, Berlin.

Michalski, R.S., Carbonell, J.G., and Mitchell, T.M. (1986), "Machine Learning, An Artificial Intelligence Approach," Vol. 2, Morgan Kaufmann Publishers Inc., San Mateo.

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

Mukouchi, Y. (1994), Inductive inference of recursive concepts, Ph.D. Thesis, RIFIS, Kyushu University 33, RIFIS-TR-CS-82, March 25th.

Natarajan, B.K. (1991), "Machine Learning, A Theoretical Approach," Morgan Kaufmann Publishers, Inc., San Mateo.

Nix, R.P. (1983), Editing by examples, Yale University, Dept. Computer Science, Technical Report 280.

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

Pitt, L., and Valiant, L.G. (1988), Computational limitations on learning from examples, *Journal of the ACM* **35**, 965 – 984.

Popper, K. (1968), "The Logic of Scientific Discovery," Harper Torch Books.

Rivest, R., Haussler, D., and Warmuth, M.K. (Eds.) (1989), Proceedings of the 2nd Annual Workshop on Computational Learning Theory, August 1989, Santa Cruz, Morgan Kaufmann Publishers Inc., San Mateo.

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

Shinohara, T. (1982), Polynomial time inference of extended regular pattern languages, *in* "Proceedings RIMS Symposia on Software Science and Engineering," Kyoto, Lecture Notes in Computer Science 147, pp. 115 – 127, Springer-Verlag, Berlin.

Solomonoff, R. (1964), A formal theory of inductive inference, *Information and Control* **7**, 1 – 22, 234 – 254.

Trakhtenbrot, B.A., and Barzdin, Ya.M. (1970) "Конечные Автоматы (Поведение и Синтез)," Наука, Москва,
English translation: "Finite Automata–Behavior and Synthesis, Fundamental Studies in Computer Science 1," North-Holland, Amsterdam, 1973.

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

Wexler, K., and Culicover, P. (1980), "Formal Principles of Language Acquisition," MIT Press, Cambridge, Massachusetts.

Wiehagen, R. (1976), Limes–Erkennung rekursiver Funktionen durch spezielle Strategien, *Journal of Information Processing and Cybernetics (EIK)*, **12**, 93 – 99.

Wiehagen, R. (1977), Identification of formal languages, *in* "Proceedings Mathematical Foundations of Computer Science, Tatranska Lomnica," (J. Gruska, Ed.), Lecture Notes in Computer Science 53, pp. 571 – 579, Springer-Verlag, Berlin.

Wiehagen, R. (1978), Characterization problems in the theory of inductive inference, *in* "Proceedings 5th Colloquium on Automata, Languages and Programming," (G. Ausiello and C. Böhm, Eds.), Lecture Notes in Computer Science 62, pp. 494 – 508, Springer-Verlag, Berlin.

Wiehagen, R. (1991), 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 543, pp. 184 – 207, Springer-Verlag, Berlin.

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

Wiehagen, R., and Zeugmann, T. (1994), Learning and Consistency, this volume.

Zeugmann, T., and Lange, S. (1995), A guided tour across the boundaries of learning recursive languages, *in* "Algorithmic Learning for Knowledge-Based Systems" (K.P. Jantke and S. Lange, Eds.), Lecture Notes in Artificial Intelligence, Vol. 961, pp. 193 – 262, Springer-Verlag.

Zeugmann, T., Lange, S., and Kapur, S. (1995), Characterizations of monotonic and dual monotonic language learning, *Information and Computation* **120**, No. 2, 1995, 155 – 173.