Characterizations of Monotonic and Dual Monotonic Language Learning

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

Steffen Lange
FB Mathematik und Informatik
Hochschule für Technik, Wirtschaft und Kultur Leipzig
Postfach 66
04251 Leipzig
E-mail: steffen@informatik.th-leipzig.de

Shyam Kapur
Department of Computer Science
James Cook University of North Queensland
Townsville, Queensland 4811
Australia
E-mail: kapur@coral.cs.jcu.edu.au

Abstract

The present paper deals with monotonic and dual monotonic language learning from positive as well as from positive and negative examples. The three notions of monotonicity reflect different formalizations of the requirement that the learner has to produce better and better generalizations when fed more and more data on the concept to be learned.

The three versions of dual monotonicity describe the concept that the inference device has to produce specializations that fit better and better to the target language. We characterize strong-monotonic, monotonic, weak-monotonic, dual strong-monotonic, dual monotonic, and monotonic & dual monotonic as well as finite language learning from positive data in terms of recursively generable finite sets. These characterizations provide a unifying framework for learning from positive data under the various monotonicity constraints. Moreover, they yield additional insight into the problem of what a natural learning algorithm should look like.

1. Introduction

The process of hypothesizing a general rule from eventually incomplete data is called inductive inference. Many philosophers of science have focused their attention on problems in inductive inference. Some of the principles developed are very much alive in *algorithmic learning theory*, a rapidly emerging science that started with the seminal papers of Solomonoff (1964) and of Gold (1967). The state of the art is excellently surveyed by Angluin and Smith (1983, 1987). For more information concerning recent developments in inductive inference, the reader is referred to the annual Workshops on Computational Learning Theory, COLT (cf., e.g., Rivest *et al.* (1989), Fulk and Case (1990) and 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)).

The present paper deals with formal language learning in the limit. In this field, many interesting and sometimes surprising results have been obtained within the last decades (cf., eg., Osherson *et al.* (1986), Case (1988), Fulk (1990), Shinohara (1990)). The general situation investigated in language learning in the limit can be described as follows: Given more and more information concerning the language to be learned, the inference device has to produce, from time to time, a hypothesis about the phenomenon to be inferred. The set of all admissible hypotheses is called the *space of hypotheses*, or, synonymously, the *hypothesis space*. The information given may contain only *positive examples*, i.e., eventually all the strings contained in the language to be learned, or both *positive and negative examples*, i.e., arbitrary strings over the underlying alphabet which are classified with respect to their containment in the unknown language. The sequence of hypotheses has to converge to a hypothesis correctly describing the object to be learned. There are many possible requirements on the sequence of hypotheses actually created. Consider the following scenario: A potential user of a learning system might prefer a particular representation of all the objects to be learned. For example,

let $\Sigma = \{a, b, ..\}$ be any non-empty finite alphabet. 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. Pattern languages were introduced by Angluin (1980a). Nix (1983) outlined interesting applications of pattern inference algorithms. Recently, Kearns and Pitt (1989) studied the learnability of pattern languages in the PAC–learning model, and Lange and Wiehagen (1991) dealt with polynomial time learnability of all pattern languages.

Consequently, looking at the potential applications, a user of a learning system might be highly interested in getting just patterns as hypotheses. On the other hand, there are many other possible representations of pattern languages, e.g., by context-sensitive grammars. Hence, it is only natural to ask whether the demand of our hypothetical user seriously affects the capabilities of learning systems. For the sake of presentation, we introduce some notations. Let $\mathcal{L} = L_0, L_1, ...$ be any family of target languages, i.e.; we are given a particular enumeration of $\mathcal{L}$ and a particular description for every $L \in \mathcal{L}$. If the family $\mathcal{L}$ is inferable with respect to the hypothesis space $\mathcal{L}$ then $\mathcal{L}$ is said to be *exactly* learnable. Furthermore, we call a target family $\mathcal{L}$ learnable by a *class preserving* learning algorithm, if there is a hypothesis space $\mathcal{G} = G_0, G_1, ...$ such that every G_j describes a language that is contained in $\mathcal{L}$, and the learning algorithm infers $\mathcal{L}$ with respect to $\mathcal{G}$. In particular, every $L \in \mathcal{L}$ has to possess a description G_j in $\mathcal{G}$, since otherwise no learning algorithm can infer $\mathcal{L}$ with respect to $\mathcal{G}$. This means that 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 $L \in \mathcal{L}$. However, we demand that *all* the hypotheses produced describe a language that is contained in the family of all target languages. At first glance, this requirement seems to be a very natural one, since any hypothesis not fulfilling it cannot be correct. Nevertheless, as previously obtained results show, at the least the efficiency of learning may be seriously affected if one insists on class preserving learning (cf., e.g., Pitt and Valiant (1988), Blum and Singh (1990), Lange and Zeugmann (1993b)). Therefore, we sometimes consider *class comprising* learning, too. In this setting a learning algorithm is allowed to use any hypothesis space $\mathcal{G} = G_0, G_1, ...$ such that any $L \in \mathcal{L}$ possesses a description G_j but $\mathcal{G}$ may additionally contain elements G_k describing languages from outside $\mathcal{L}$.

The *main goal* of this paper consists in partially answering the question of what a *natural learning algorithm* should look like. We believe that the answer to this question should be related to the classes of learning problems one wants to deal with. The starting point of our investigations goes back to different learning strategies that have controversially been discussed 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 till it eventually reached 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 extensive debate in the machine learning community for and against each of these learning modes (cf., e.g., Michalski *et al.* (1984, 1986) or Kodratoff and Michalski (1990)). Looking for mathematically sound formalizations of the concept of generalization, Jantke (1991a, 1991b) discovered an interesting connection to the problem whether or not a learning algorithm has to use non-monotonic reasoning in order to learn. Subsequently, Wiehagen (1991) refined Jantke's (1991a) approach, and Kapur (1992) introduced the dual version 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 guessed later than L_i (cf. Definitions 4 and 6, part (A)).

Wiehagen (1991) proposed to interpret "better generalization" with respect to the language L having to be identified. Consequently, we require $L_i \cap L \subseteq L_j \cap L$ in case L_j appears later in the sequence of guesses than L_i does (cf. Definition 4 (B)). This means that a new hypothesis is never allowed to reject some string that a previously generated guess already *correctly includes*.

On the other hand, it is only natural to consider the dual version of the latter requirement as well. Intuitively, dual monotonicity describes the following requirement: If, at any stage, the learner outputs a hypothesis *correctly excluding* a string s from the language to be learned, then any subsequent guess must also exclude that string (cf. Definition 6 (B)).

The third version of monotonicity, which we call weak-monotonicity and dual weak-monotonicity, respectively, is derived from non-monotonic logic. In classical logic it is required that more assumptions do not lead to a decrease of the set of derivable theorems. This requirement is the so-called monotonicity principle. If we interpret the data that a learning algorithm is fed as assumptions about the target language then we directly see that strong-monotonic inference is just the learning model that realizes this classical approach. On the other hand, non-monotonic logic aims to model common sense that cannot be formalized in classical logic. Hence, it is only natural to ask whether or not non-monotonic reasoning has to be taken into account in the design of learning algorithms. A natural weakening of the monotonicity principle is cumulativity (cf., e.g., Brewka (1991)). In a cumulative logic the monotonicity principle has to be fulfilled as long as new assumptions added to the set of assumptions do not contradict the theorems derivable from it. Transforming cumulativity and its dual analogue to learning results in requiring $L_i \subseteq L_j$ ($L_i \supseteq L_j$) as long as there are no data fed to the inference device after having produced L_i that contradict L_i (cf. Definitions 4 and 6, part (C)).

In the following we restrict ourselves to deal exclusively with the learnability of indexed families of non-empty uniformly recursive languages. This case is of special interest with respect to potential applications. The first problem arising naturally is to relate the learning power of all types of monotonic language learning to one another as well as

to previously studied modes of inference. This question has been completely answered in Lange *et al.* (1992) and in Lange and Zeugmann (1993c). In this paper, we mainly concentrate ourselves to study what all the described learning models have in *common* and what their *differences* are. It turns out that *characterizations* are a very useful method of achieving this goal. In Lange and Zeugmann (1994), we presented characterization theorems for all types of class preserving monotonic and dual monotonic language learning from positive and negative data. In particular, we showed that there is essentially one learning algorithm which can perform any of the inference tasks described above. This insight has been achieved by a suitable modification of Angluin's (1980b) characterization of learning in the limit from positive data in terms of finite and non-empty tell-tale sets. In this paper, we will present a *unifying framework for learning from positive data* under the various monotonicity constraints. Although the situation is much more subtle than in the case of learning from positive and negative data, the overall insight remains the same. In essence, there is one learning algorithm that may be applied to all inference tasks. The particular differences between the different models of monotonic and dual monotonic language learning may be described mainly by the specific properties of the relevant finite and non-empty tell-tale sets. As we shall see, our characterization theorems may successfully be applied to deciding what the relevant "natural learning algorithms" should look like. Since within the current level of precision nothing more satisfactory can be added, we will discuss these issues in more detail throughout the paper.

The paper is structured as follows. Section 2 presents preliminaries, i.e., notations and definitions. The characterization theorems are established in Section 3 and the subsections therein. Moreover, Subsection 3.4 is devoted to applications of our characterization theorems. Finally, in Section 4 we discuss the results obtained and outline open problems. All references are given in Section 5.

2. Preliminaries

Let $\mathbb{N} = \{0, 1, 2, ...\}$ be the set of all natural numbers. By $c : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ we denote ***Cantor's pairing function***, i.e., $c(x, y) = ((x + y)^2 + 3x + y)/2$ 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. 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 $\text{range}(t) = \{s_k \mid k \in \mathbb{N}\} = L$. Then t is said to be a ***text*** for L or, synonymously, a ***positive presentation***. Furthermore, let $i = (s_0, b_0), (s_1, b_1), ...$ be an infinite sequence of elements of $\Sigma^* \times \{+, -\}$ such that $\text{range}(i) = \{s_k \mid k \in \mathbb{N}\} = \Sigma^*$, $i^+ = \{s_k \mid (s_k, b_k) = (s_k, +), k \in \mathbb{N}\} = L$ 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***. 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 dealing 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

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

As an example, we consider the set $\mathcal{L}$ of all context-sensitive languages over Σ. $\mathcal{L}$ may be regarded as an indexed family of uniformly recursive languages (cf. Hopcroft and Ullman (1969)). 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 string, or first outputs a hypothesis, i.e., a number encoding a certain computer program, and then it requests the next input string (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 this 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 the hypothesis space an enumerable family of grammars $G_0, G_1, G_2, ...$ over the terminal alphabet Σ satisfying $\mathcal{L} \subseteq \{L(G_j) | j \in \mathbb{N}\}$. Moreover, we require that membership in $L(G_j)$ be 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. When an IIM outputs a number j, we interpret it to mean that the machine is hypothesizing the grammar G_j.

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, $L \in \mathcal{L}$, 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 $\mathcal{G}$ if and only if, 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 up to the (unknown) point of convergence, whenever an IIM identifies the language L, some form of learning must have taken place. For this reason, from here on the terms *infer*, *learn*, and *identify* are used interchangeably.

In Definition 1, 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. This means that 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($\mathcal{L}$) $= \{L(G_j) | j \in \mathbb{N}\}$. 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. We adopt this convention in the definitions of the learning types below.

The following proposition clarifies the relations between exact, class preserving, and class comprising learning in the limit.

Proposition 1. (Lange and Zeugmann, 1993b)

(1) $ELIM-TXT = LIM-TXT = CLIM-TXT.$

(2) $ELIM-INF = LIM-INF = CLIM-INF.$

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 is required that the IIM declares that it has finished the learning task.

Definition 2. (Gold, 1967; Trakhtenbrot and Barzdin, 1970) *Let $\mathcal{L}$ be an indexed family, $L \in \mathcal{L}$, 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, $MCFIN{-}TXT$- [$CFIN{-}INF$]-identifies $\mathcal{L}$ with respect to $\mathcal{G}$ if and only if, for each $L \in \mathcal{L}$, $MCFIN{-}TXT$- [$CFIN{-}INF$]-identifies L with respect to $\mathcal{G}$.

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

Next, we want to formally define strong-monotonic, monotonic and weak-monotonic inference. But before doing this, we first define *consistent* identification. Consistently working 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 3. (Barzdin, 1974) *Let $\mathcal{L}$ be an indexed family, $L \in \mathcal{L}$, and let $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ be a hypothesis space.* ***An IIM M CCONS–TXT- [CCONS–INF]-identifies* 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, $MCCONS{-}TXT$- [$CCONS{-}INF$]-identifies $\mathcal{L}$ with respect to $\mathcal{G}$ if and only if, for each $L \in \mathcal{L}$, $MCCONS{-}TXT$- [$CCONS{-}INF$]-identifies L with respect to $\mathcal{G}$.

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

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

Definition 4. (Jantke, 1991a, Wiehagen, 1991) *Let $\mathcal{L}$ be an indexed family, $L \in \mathcal{L}$, 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) ***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 has to work 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 collections of all those indexed families $\mathcal{L}$ for which there is 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.

Note that even $SMON{-}TXT$ contains interesting "natural" families of formal languages. For example, let $DPAT$ be the family of all languages L for which there are finitely many patterns $p_{i_1}, ..., p_{i_k}$ such that $L = \bigcap_{j=1}^{k} L(p_{i_j})$. Then there exist a class preserving hypothesis space $\mathcal{G}$ and an IIM M such that M strong-monotonically identifies $DPAT$ with respect to $\mathcal{G}$ (cf. Lange and Zeugmann (1992, 1993a)).

Figure 1 summarizes 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 conveys 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 (or path) from vertex A to vertex B indicates that A is a proper subset of B, a bidirectional edge represents $A = B$, and no edge (or path) between these vertices implies that A and B are incomparable.

Monotonic Learning from Text versus Monotonic Learning from Informant

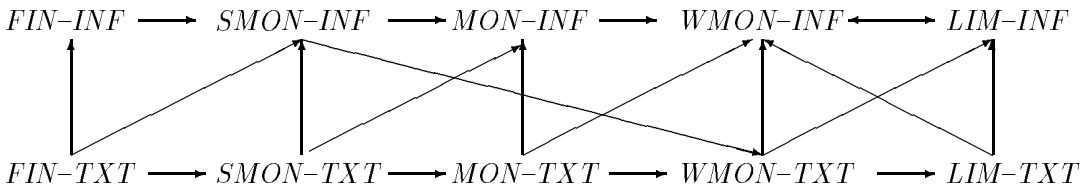

Figure 1

Next, we define *conservative* IIMs. Intuitively speaking, conservative IIMs maintain their actual hypotheses at least as long as they have not seen data contradicting them.

Hence, whenever a conservative IIM performs a mind change it is because it has perceived clear inconsistency between its guess and the input.

Definition 5. (Angluin, 1980b) *Let $\mathcal{L}$ be an indexed family, $L \in \mathcal{L}$, 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 [from 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} [$L(G_{j_x})$ either must fail to contain some string $s \in i^+_{x+k}$ or must generate 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 collections of sets *CCONSERVATIVE–TXT* and *CCONSERVATIVE–INF* are defined in a manner analogous to that above.

Note that $\lambda WMON-TXT = \lambda CONSERVATIVE–TXT$ as well as $\lambda WMON-INF = \lambda CONSERVATIVE–INF$ for all $\lambda \in \{C, \varepsilon, E\}$, where ε denotes the empty string (cf. Lange and Zeugmann (1993a)).

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

Definition 6. *Let $\mathcal{L}$ be an indexed family, $L \in \mathcal{L}$, 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 $CSMON^d$–TXT, $CSMON^d$–INF, $CMON^d$–TXT, $CMON^d$–INF, $CWMON^d$–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 or 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 *et al.* (1992), and Lange and Zeugmann (1994)). The semantics of Figure 2 is analogous to that of Figure 1. On comparison 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

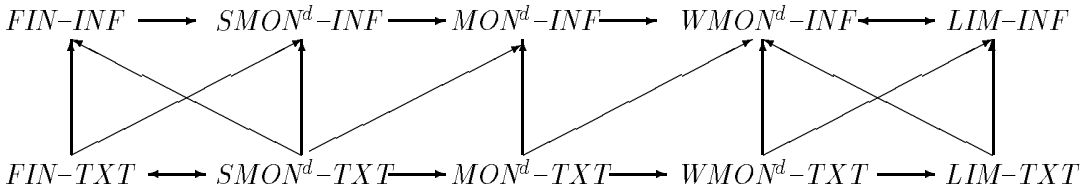

Figure 2

Note that the notions of monotonicity and of dual monotonicity are truly duals of *each other*. Next, we combine the monotonicity constraints from Definition 4 and Definition 6. As we shall see, this helps us to gain a better understanding of the relationships between monotonic inference of languages and other well-known types of language learning.

Definition 7. *Let $CSMON^{\&}-TXT$ [$CSMON^{\&}-INF$] denote the class of indexed families learnable from text [informant] by an IIM that works strong-monotonically as well as dual strong-monotonically. The learning types $CMON^{\&}$–TXT, $CMON^{\&}$–INF, $CWMON^{\&}-TXT$ and $CWMON^{\&}-INF$ are defined analogously.*

Finally, we present two figures relating all types of class preserving monotonic language learning to all types of dual monotonic inference of indexed families. For the sake of readability, we separate the results for language learning from text (Fig. 3) from those ones dealing with language learning from positive and negative data (Fig. 4). Again, we restrict ourselves to the class preserving case, since it reflects the typical relations between the introduced learning types. Note that missing edges between learning types not connected by a path in the underlying graph are incomparable. Additional results which give further insights into the relations between the defined learning types can be found in Lange *et al.* (1992) as well as in Lange and Zeugmann (1993c, 1994).

Monotonic and Dual Monotonic Learning from Text

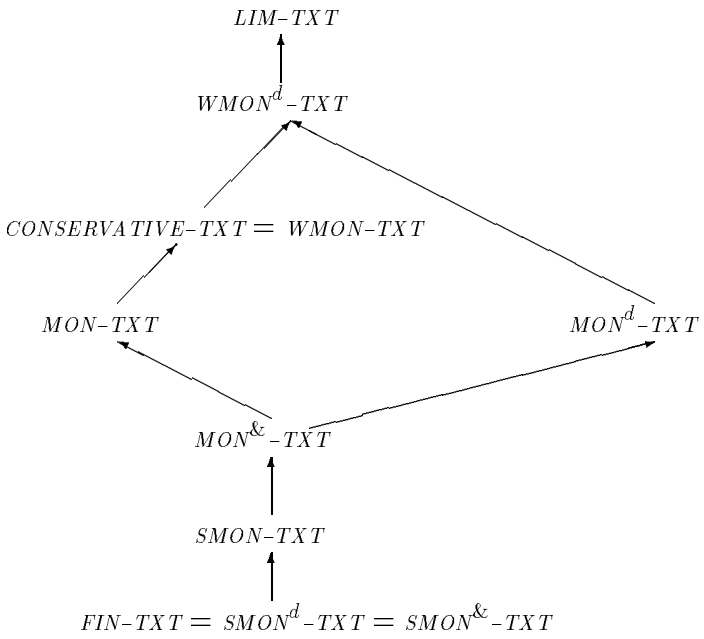

Figure 3

The next figure relates monotonic and dual monotonic language learning on informant.

Monotonic versus Dual Monotonic Inference from Informant

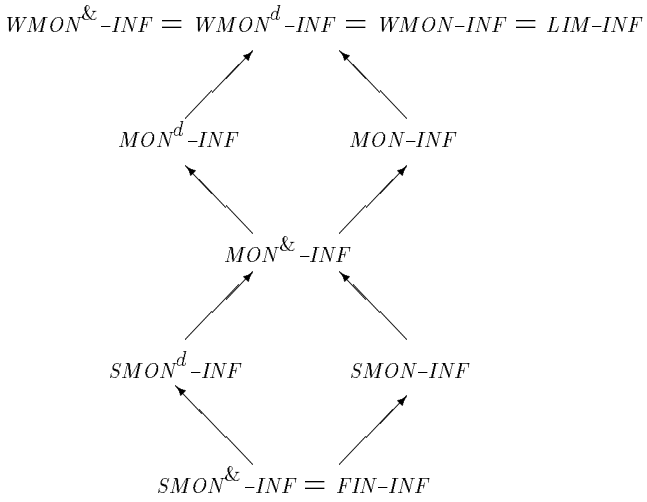

Figure 4

3. Characterization Theorems

In this section, we present characterizations of all types of monotonic and of dual monotonic language learning from positive data. Characterizations play an important role in that they lead to deeper insights into how algorithms performing the inference process may work (cf., e.g., Blum and Blum (1975), Wiehagen (1977, 1991), Angluin (1980b), Zeugmann (1983), Jain and Sharma (1989)). Starting with the pioneering paper of Blum and Blum (1975), several theoretical frameworks have been used for characterizing learning types. For example, characterizations in inductive inference of recursive functions have been formulated in terms of complexity theory (cf. Blum and Blum (1975), Wiehagen and Liepe (1976), Zeugmann (1983)) and in terms of computable numberings (cf., e.g., Wiehagen (1977, 1991) and the references therein). Surprisingly, some of the mentioned characterizations have been successfully applied to solve non-trivial problems in abstract complexity theory. For example, by characterizing the reliably identifiable function classes in two quite different ways, Blum and Blum (1975) proved that the operator-honest recursive functions are precisely the everywhere operator-compressed recursive functions. Other examples of techniques par solving problems in abstract complexity theory via the characterization of inferable function classes are outlined in Zeugmann (1983, 1988). Moreover, how to solve the same problems without using these characterizations remains open. It seems that characterizations help to gain a deeper understanding of the theoretical framework the concepts for characterizing learning types are borrowed from. Furthermore, characterizations may help us to gain a better understanding of the properties objects should have in order for them to be inferable in the desired sense. A very illustrative example is Angluin's (1980b) characterization of those indexed families for which learning in the limit from positive data is possible. In particular, this theorem provides insight into the problem how to deal with overgeneralizations. Theorem 17 below offers an alternative way to resolve this question.

Proposition 2. (Angluin, 1980b) *Let $\mathcal{L}$ be an indexed family of recursive languages. Then $\mathcal{L} \in LIM{-}TXT$ if and only if there is an effective procedure which on any 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 $L_z \not\subset L_j$.*

Originally, this theorem characterized all those indexed families $\mathcal{L}$ of recursive languages which are inferable with respect to $\mathcal{L}$. However, a straightforward application of Proposition 1 yields that Proposition 2 completely characterizes indexed families which are inferable in the limit from positive data.

Although Angluin (1980b) established a sufficient condition that guarantee exact conservative learning from positive data, it remained open whether *ECONSERVATIVE–TXT* may be characterized in terms of finite non-empty sets. In the following subsection we present a solution to this longstanding open problem. However, when dealing with

conservative learning the situation turns out to be more subtle than in the case of learning in the limit. The new feature we have to take into account is that conservative learning is sensitive to the particular choice of the hypothesis space. Remember that $\lambda WMON{-}TXT = \lambda CONSERVATIVE{-}TXT$ for all $\lambda \in \{C, \varepsilon, E\}$. As a matter of fact, we have the following proposition.

Proposition 3. (Lange and Zeugmann, 1993c)

$$EWMON{-}TXT \subset WMON{-}TXT \subset CWMON{-}TXT \subset LIM{-}TXT.$$

As a consequence, in characterizing class preserving and class comprising conservative learning one has to construct an appropriate hypothesis space. Furthermore, we combine this construction with an effective procedure generating recursive tell-tale sets rather than recursively enumerable ones as in Proposition 2. Finally, we state the announced theorems as characterizations of weak-monotonic language learning from positive data.

3.1. The Characterization of Weak-Monotonic Inference

In this subsection we characterize exact, class preserving, and class comprising weak-monotonic inference. For the sake of readability, we start with the class preserving case, since it best represents the developed proof technique. As already mentioned, we characterize $WMON{-}TXT$ in terms of recursively generable finite tell-tales. 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 input j, generates all elements of T_j and stops. If the computation of $g(j)$ stops and there is no output, then T_j is considered to be empty. Finally, for notational convenience, we use $L(\mathcal{G})$ to denote $\{L(G_j) \mid j \in \mathbb{N}\}$ for any space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ of hypotheses.

Theorem 1. *Let $\mathcal{L}$ be an indexed family. Then $\mathcal{L} \in WMON{-}TXT$ if and only if there are a space of hypotheses $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ and a recursively generable family $(T_j)_{j\in\mathbb{N}}$ of finite and non-empty sets such that*

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

(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 $\mathcal{L} \in WMON{-}TXT = CONSERVATIVE{-}TXT$. Then there are an IIM M and a hypothesis space $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ such that M infers any $L \in \mathcal{L}$ conservatively with respect to $\hat{\mathcal{G}}$. We proceed in showing how to construct $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$. This is done in two steps. First, we construct a space of hypotheses $\tilde{\mathcal{G}} = (\tilde{G}_j)_{j\in\mathbb{N}}$ as well as a 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 $\mathcal{G}$. Let c be Cantor's pairing function. We define the hypothesis space $\tilde{\mathcal{G}}$ as well as the wanted family $(\tilde{T}_j)_{j\in\mathbb{N}}$ as follows: On input j compute $k, x \in \mathbb{N}$ such that $j = c(k, x)$. Then set $\tilde{G}_{c(k,x)} = \hat{G}_k$. Furthermore, for any language $L(\hat{G}_k)$, let t^k be the canonically ordered text of $L(\hat{G}_k)$.

We define:

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

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

The intuitive idea behind the definition of the family $(\tilde{T}_j)_{j \in \mathbb{N}}$ is as follows. If an IIM M is fed initial segments of a text then, in general, by looking at the outputs, there is almost no hope of determining what the IIM M has learned from its input. However, in the particular case where we have an index k of the language L, the canonical text t^k of which is successively fed M, there is a syntactical criterion that can be tested effectively. Namely, if $M(t_z^k) = k$, then we know for sure that M has done a pretty good job on input t_z^k. Clearly, it has output a correct guess for the language it should learn. Therefore, it seems very reasonable to collect all these initial segments in the family $(\tilde{T}_j)_{j \in \mathbb{N}}$. But we can do even better, i.e., it suffices to collect the ranges of all these initial segments.

The desired space of hypotheses $\mathcal{G}$ is obtained from $\tilde{\mathcal{G}}$ by simply striking off all grammars $\tilde{G}_{c(k,x)}$ for which $\tilde{T}_{c(k,x)} = \emptyset$. 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 preserve notational convenience, we refer to T_j as to $T_{c(k,x)}$, 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 $j = c(k,x)$ we have $L = L(G_{c(k,x)})$. For this purpose, due to our construction, it suffices to show that $\tilde{T}_{c(k,x)} \neq \emptyset$. 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, $\tilde{T}_{c(k,y)} = \text{range}(t_y)$. Hence, by the convention made above, we get that $T_{c(k,y)} = \text{range}(t_y)$. Moreover, it immediately follows that $L = L(G_{c(k,x)})$ for any $x \geq y$. This proves Property (1).

Finally, we have to show (3). It results from the requirement that any conservatively working IIM is never allowed to output an overgeneralized hypothesis, i.e., a guess that generates a proper superset of the language to be inferred. To see this, suppose the converse, i.e., that 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 $j = c(k,x)$. Let $s_0, ..., s_y$ be the sequence of strings of T_j in canonical order with respect to $L(G_{c(k,x)})$ 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 $T_j \subseteq L(G_z) \subset L(G_j) = L(G_{c(k,x)})$. Finally, M has to infer $L(G_z)$ on 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 $\mathcal{G}$. So let $L \in \mathcal{L}$ and let t be any text for L, and let $x \in \mathbb{N}$.

$M(t_x) =$ "If $x = 0$ or $x > 0$ and M when successively fed t_{x-1} does not output a

hypothesis, then goto (B). Otherwise, goto (A).

(A) Let $j = M(t_{x-1})$. Test whether $t_x^+ \subseteq L(G_j)$. If it is, output j and request the next input. Otherwise, goto (B).

(B) For $j = 0, ..., x$, generate T_j and test whether $T_j \subseteq t_x^+ \subseteq L(G_j)$. If there exists at least one j fulfilling the test, output the minimal one. Otherwise, output nothing and request the next input."

Since all of the 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. Since the machine M changes its mind only if it finds an inconsistency in (A), it works conservatively.

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

Let z be the least k such that $L = L(G_k)$. Consider T_0, ..., T_z. Then there must be an $x \geq z$ such that $T_z \subseteq t_x^+ \subseteq L(G_z)$. That means, at least after M has been fed t_x, the machine M always outputs a hypothesis and it never outputs a guess $j > x$ because $z \in \{k \leq x \,|\, T_k \subseteq t_x^+ \subseteq L(G_k)\}$. Moreover, since M changes its mind if and only if it receives some string that misclassifies its current guess, we see that any rejected hypothesis is never repeated in some subsequent step. Finally, since z can never be rejected, M has to converge.

Claim 2. If M converges, say to j, then $L = L(G_j)$.

Suppose the converse, i.e., M converges to j and $L \neq L(G_j)$.

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

Consequently, there is at least one string $s \in L \setminus L(G_j)$ that has to appear sometime in t, say in t_r for some r. Thus, $t_r^+ \not\subseteq L(G_j)$. Hence, after seeing t_r, our IIM M never outputs j, a contradiction.

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

Then, we may restrict ourselves to the case $L \subset L(G_j)$, since otherwise we are again in Case 1. On the other hand, due to the definition of M there should be an $r \in \mathbb{N}$ such that M in (B) verifies $T_j \subseteq t_r^+ \subseteq L(G_j)$, since otherwise it cannot output j at least once. Moreover, since $L = L(G_z)$ and $t_r^+ \subseteq L(G_z)$ for any $r \in \mathbb{N}$, we conclude $T_j \subseteq L(G_z)$. Hence $L = L(G_z) \not\subset L(G_j)$ by Property (3), a contradiction. Q.E.D.

Because $WMON^{\&} - TXT = WMON - TXT$, the theorem above yields a characterization of $WMON^{\&} - TXT$, too.

Kapur and Bilardi (1992) also established a characterization of class preserving conservative learning. Their main characterization differs at least conceptually from the one presented above. In order to see the difference, we need the following notion. Let A be a finite set and let $\mathcal{L}$ be an indexed family. A language $L \in \mathcal{L}$ is said to be a least upper bound of A iff $A \subseteq L$ and any language $\hat{L} \in \mathcal{L}$ containing A is not a proper subset of L. Kapur and Bilardi (1992) showed that conservative learning is equivalent to the existence of a recursive enumeration of pairs of finite sets and grammars such that, in each pair, the language corresponding to the grammar is a least upper bound of the corresponding

finite set, and, for each $L \in \mathcal{L}$, there is at least a corresponding pair. Consequently, this characterization is conceptually based on the judicious use of a function computing least upper bounds. Our approach is in some sense the converse in that we construct a suitable enumeration $\hat{\mathcal{L}}$ of $\mathcal{L}$ and for every language $\hat{L} \in \hat{\mathcal{L}}$ a recursive and finite set $\hat{T}$ such that $\hat{L}$ is a least upper bound of $\hat{T}$.

The next theorem establishes the characterization of exact weak-monotonic language learning from positive data. Hence, it solves the problem that remained open in Angluin (1980b).

Theorem 2. *Let $\mathcal{L}$ be an indexed family. Then $\mathcal{L} \in EWMON - TXT$ 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 $L_z \not\subset L_j$.*

Proof. The proof is very similar to that of Theorem 1. The only difference is that one has to use the indexed family $\mathcal{L}$ itself as hypothesis space. Suppose there is a language $L \in \mathcal{L}$ that appears more than once in the enumeration of $\mathcal{L}$. Then it might well happen that there are indices of L that do not appear in the sequence of hypotheses produced by M when fed the canonical text of L. Therefore, $(T_j^y)_{j,y\in\mathbb{N}}$ may contain empty sets, too. On the other hand, every language is learned by M. Hence, for every L there has to be an index j such that $T_j^y \neq \emptyset$ for all but finitely many $y \in \mathbb{N}$. The rest is proved *mutatis mutandis* as in the demonstration of Theorem 1. For a complete proof, the reader is referred to Lange and Zeugmann (1993e). Q.E.D.

Finally, we consider class comprising weak-monotonic learning. Although the following theorem looks very similar to Theorem 1 the proof has to handle a new problem. This is caused by the fact that *a priori* it is hard to say how an IIM behaves on languages that do not belong to the given indexed family. In particular, it is nowhere required that an IIM fulfills the weak-monotonicity constraint when fed a text of a language $L \notin \mathcal{L}$. However, we have found a method of overcoming these difficulties. On the other hand, it seems that our developed technique does not generalize to handle, for example, class comprising strong monotonic inference. Therefore, we skip the proof here, and refer the interested reader to Lange and Zeugmann (1993d).

Theorem 3. *Let $\mathcal{L}$ be an indexed family. Then $\mathcal{L} \in CWMON - TXT$ 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 non-empty sets such that*

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

(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 $L(G_z) \not\subset L(G_j)$.*

3.2. The Characterization of Strong-Monotonic Inference

This subsection deals with the characterization of strong-monotonic inference. Again, we have to take into account that the learning power of strong-monotonic inference depends on the allowed classes of hypothesis spaces.

Proposition 4. (Lange and Zeugmann, 1993c)

$$ESMON-TXT \subset SMON-TXT \subset CSMON-TXT.$$

In the following we present characterizations of all types of strong-monotonic inference. As it turned out, the proof technique presented above applies *mutatis mutandis* to the following theorem.

Theorem 4. *Let $\mathcal{L}$ be an indexed family. Then $\mathcal{L} \in SMON-TXT$ if and only if there are a space of hypotheses $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ and a recursively generable family $(T_j)_{j\in\mathbb{N}}$ of finite and non-empty sets such that*

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

(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 $L(G_j) \subseteq L(G_z)$.*

Proof. Necessity: Let M be any IIM witnessing $\mathcal{L} \in SMON-TXT$ w.r.t. some hypothesis space $\hat{\mathcal{G}}$. The hypothesis space $\mathcal{G}$ and the uniformly recursively generable family $(T_j)_{j\in\mathbb{N}}$ of finite and non-empty sets are analogously defined as in in the proof of Theorem 1. Using the same arguments as above, one immediately obtains Properties (1) and (2). In order to prove (3), let $T_j \subseteq L(G_z)$. We have to show that $L(G_j) \subseteq L(G_z)$. Let k, x be the uniquely determined numbers with $j = c(k,x)$. Furthermore, let $s_0, ..., s_y$ be the strings of T_j in canonical order with respect to $L(G_{c(k,x)})$ such that $M(s_0, ..., s_y) = k$ for the first time. Since $T_j \subseteq L(G_z)$, we see that $s_0, ..., s_y$ is also an initial segment of some text for $L(G_z)$. Consequently, $s_0, ..., s_y$ may be extended to a text for $L(G_z)$. Finally, since M has to infer $L(G_z)$ too, there should be an $n \in \mathbb{N}$ and a finite extension σ of strings of $L(G_z)$ such that $M(s_0, ..., s_y, \sigma) = n$ and $L(G_z) = L(\hat{G}_n)$. M works strong-monotonically and hence, by the transitivity of "$\subseteq$," we obtain $L(G_j) \subseteq L(G_z)$.

Sufficiency: It suffices to show that there is an IIM M that strong-monotonically identifies $\mathcal{L}$ from text with respect to $\mathcal{G}$. Let $L \in \mathcal{L}$, let t be any text for L, and let $x \in \mathbb{N}$. The wanted IIM M is defined as follows:

$M(t_x) =$ "Generate T_j and test whether $T_j \subseteq t_x^+ \subseteq L(G_j)$ for $j = 0, ..., x$.
In case there exists one j fulfilling the test, output the minimal one and request the next input.
Otherwise output nothing and request the next input."

Now, it is not hard to prove that M infers $\mathcal{L}$ strong-monotonically with respect to $\mathcal{G}$. We omit the details. Q.E.D.

Next, we present a characterization of $ESMON-TXT$.

Theorem 5. *Let $\mathcal{L}$ be an indexed family. Then $\mathcal{L} \in ESMON-TXT$ 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 $L_j \subseteq L_z$.*

Proof. The proof is a straightforward modification of the demonstration of Theorem 4 and therefore omitted. Q.E.D.

Despite the simplicity of their proofs, the theorems presented in this subsection allow several interesting applications which we present in Subsection 3.4.

Finally, we characterize class comprising strong-monotonic inference. The main problem we have to deal with is similar to the one we faced when characterizing class comprising weak-monotonic learning. Proposition 4 tells us that every IIM inferring an indexed family $\mathcal{L} \in CSMON-TXT \setminus SMON-TXT$ has to output hypotheses which do not describe a language from $\mathcal{L}$. Therefore, our standard tell-tale construction provides only information concerning languages that belong to $\mathcal{L}$. Hence, we may try to refine the tell-tale definition as we did in the proof of Theorem 3. But this is easier said than done. Alternatively, we could try to develop a new technique to handle the sufficiency part. As it turned out, the latter approach succeeded. As a consequence, the following characterization has an interesting feature distinguishing it from the theorems presented above. All the previous characterizations share the following structure. They establish necessary and sufficient conditions in terms of a suitable hypothesis space $\mathcal{G}$ and a tell-tale family that is defined with respect to it. Moreover, in the sufficiency part we always proved inferability with respect to $\mathcal{G}$ provided the stated conditions are fulfilled. The following characterization differs from the previously obtained ones, since it yields a slightly weaker conclusion. Assuming that the stated conditions are satisfied we may only prove that there exists a suitable hypothesis space with respect to which the indexed family under consideration is strong-monotonically learnable.

Theorem 6. *Let $\mathcal{L}$ be an indexed family. Then $\mathcal{L} \in CSMON-TXT$ 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) $\mathrm{range}(\mathcal{L}) \subseteq L(\mathcal{G})$,

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

(3) *for all $L \in \mathcal{L}$ and all $j \in \mathbb{N}$, if $T_j \subseteq L$ then $L(G_j) \subseteq L$.*

Proof. The necessity can be proved in the same way as in Theorem 4, since Assertion (3) deals exclusively with languages that belong to $\mathcal{L}$.

Sufficiency: First, we define a new hypothesis space $\hat{\mathcal{G}}$. Then we prove $\mathcal{L} \in CSMON-TXT$ w.r.t. $\hat{\mathcal{G}}$. The hypothesis space $\hat{\mathcal{G}} = (\hat{G}_j)_{j \in \mathbb{N}}$ is obtained form $\mathcal{G}$ by enumerating its closure with respect to finite unions. Using an appropriate coding it is easy to see that membership in $L(\hat{G}_j)$, $j \in \mathbb{N}$, remains uniformly decidable. Next, we define the desired IIM. Let $L \in \mathcal{L}$, let t be any text for L, and let $x \in \mathbb{N}$.

$M(t_x) =$ "If $x = 0$ or $x > 0$ and M when successively fed t_{x-1} does not output a hypothesis, then goto (A). Otherwise, goto (B).

(A) For $j = 0, ..., x$, generate T_j and test whether $T_j \subseteq t_x^+ \subseteq L(G_j)$. In case there exists one j fulfilling the test, output the canonical index of $L(G_j)$ in the hypothesis space $\hat{\mathcal{G}}$. Otherwise, output nothing and request the next input.

(B) Let $j = M(t_{x-1})$. Test whether $t_x^+ \subseteq L(\hat{G}_j)$. In case it is, output j and request the next input.
Otherwise, test for $k \leq x$ whether $T_k \subseteq t_x^+ \subseteq L(G_k)$. In case there exists one k fulfilling the test, choose the minimal one and output the canonical index of $L(G_j) \cup L(G_k)$ in the hypothesis space $\hat{\mathcal{G}}$.
Otherwise, output j and and request the next input."

We leave it to the reader to verify that M strong-monotonically learns $\mathcal{L}$ w.r.t. $\hat{\mathcal{G}}$. Q.E.D.

3.3. The Characterization of Dual Strong-Monotonic Inference

In this subsection, we present the characterization of $SMON^d-TXT$ and postpone discussion of $MON-TXT$ for a moment, since it deserves special attention. Since $SMON^d-TXT = SMON^{\&}-TXT = FIN-TXT$, we only characterize $FIN-TXT$ below. Note that a slightly weaker theorem has been obtained independently by Mukouchi (1992). The difference is caused by Mukouchi's definition of finite identification from text, which demands that any indexed family $\mathcal{L}$ be finitely inferred with respect to $\mathcal{L}$ itself. Consequently, one should ask whether or not the latter requirement might lead to a decrease in the inferring power. It does not, as we shall see.

Our next characterization has some special features distinguishing it from the characterizations previously given. As pointed out above, dealing with characterizations has been motivated by the aim to elaborate a unifying approach to monotonic inference. Concerning weak-monotonic and strong-monotonic language learning, this goal has been completely met by showing that there is essentially one algorithm, i.e., that described in the proofs of Theorem 1 and Theorem 4, respectively, which can perform the desired inference task, if the hypothesis space is appropriately chosen. The next theorem yields an even stronger implication. Namely, it shows that, if there is a class preserving hypothesis space at all such that $\mathcal{L} \in FIN-TXT$ with respect to this space, then *one can always use* $\mathcal{L}$ *itself* as a space of hypotheses, thereby again applying essentially one and the same inference procedure.

Theorem 7. *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* $L_j = L_k$.

Proof. Necessity: Let $\mathcal{L} \in FIN{-}TXT$. Then, there are a space $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ of hypotheses and an IIM M such that M finitely infers $\mathcal{L}$ with respect to $\mathcal{G}$. We show how to construct $(T_j)_{j \in \mathbb{N}}$. This is done in two steps. First, we construct $(\hat{T}_j)_{j \in \mathbb{N}}$ with respect to the space $\mathcal{G}$ of hypotheses. Then, we describe a procedure yielding the wanted family $(T_j)_{j \in \mathbb{N}}$ with respect to $\mathcal{L}$.

Let $k \in \mathbb{N}$ be arbitrarily fixed. Furthermore, let t^k be the canonically ordered text of $L(G_k)$. Since M infers $L(G_k)$ finitely on t^k, there exists an $x \in \mathbb{N}$ such that $M(t^k_x) = m$ with $L(G_k) = L(G_m)$. We set $\hat{T}_k = \text{range}(t^k_x)$. The desired family $(T_j)_{j \in \mathbb{N}}$ is obtained as follows. Let $z \in \mathbb{N}$. In order to get T_z search for the least $j \in \mathbb{N}$ such that $\hat{T}_j \subseteq L_z$. Set $T_z = \hat{T}_j$. Note that at least one wanted j has to exist, since, for any set $\hat{T}_k$, there is a text t of some language $L \in \mathcal{L}$ such that $\hat{T}_k \subseteq \text{range}(t)$.

We have to show that $(T_j)_{j \in \mathbb{N}}$ fulfills the announced properties. Due to our construction, Property (1) holds obviously. It remains to prove (2). Suppose z, $y \in \mathbb{N}$ such that $T_z \subseteq L_y$. In accordance with our construction, there is an index k such that $T_z = \hat{T}_k$. Moreover, by construction, there is an initial segment of the canonically ordered text t^k of $L(G_k)$, say t^k_x, such that $\text{range}(t^k_x) = \hat{T}_k$. Furthermore, $M(t^k_x) = m$ with $L(G_k) = L(G_m)$. Since $\hat{T}_k \subseteq L_y$, t^k_x forms an initial segment of some text for L_y. Taking into account that M finitely infers L_y from any text and that $M(t^k_x) = m$, we immediately obtain $L_y = L(G_m)$. Finally, due to the definition of T_z, we additionally know that $\hat{T}_k \subseteq L_z$, hence the same argument again applies and yields $L_z = L(G_m)$. Consequently, $L_z = L_y$. This proves (2).

Sufficiency: It suffices to prove that there is an IIM M inferring any $L \in \mathcal{L}$ finitely from any text with respect to $\mathcal{L}$. So let $L \in \mathcal{L}$, let t be any text for L, and $x \in \mathbb{N}$.

$M(t_x) =$ "Generate T_j for $j = 1, ..., x$ and test whether $T_j \subseteq t^+_x \subseteq L_j$.
In case there exists one j fulfilling the test, output the minimal one and stop.
Otherwise output nothing and request the next input."

We leave it to the reader to show that M finitely learns $\mathcal{L}$ w.r.t. $\mathcal{L}$. Q.E.D.

As a direct consequence of Theorem 7 we obtain the equality of exact and class preserving finite learning. Hence, we additionally get that $ESMON^d{-}TXT = SMON^d{-}TXT$. Moreover, a closer look at the proof of Theorem 7 immediately allows the following corollary.

Corollary 8. Let $\mathcal{L}$ be an indexed family such that $\mathcal{L} \in FIN{-}TXT$. Furthermore, let $\mathcal{G}$ be any class preserving hypothesis space for $\mathcal{L}$. Then $\mathcal{L} \in FIN{-}TXT$ with respect to $\mathcal{G}$.

Intuitively speaking, the latter corollary says that class preserving finite language learning from positive data is *invariant* with respect to the particular choice of the hypothesis space. Consequently, class preserving dual strong-monotonic inference from positive data is invariant with respect to the choice of the class preserving hypothesis space.

Interestingly enough, while this invariance remains valid for $CFIN-TXT$, it does not hold for $CSMON^d-TXT$. This is caused by the fact that $CFIN-TXT = FIN-TXT$ but $SMON^d-TXT \subset CSMON^d-TXT$ (cf. Lange and Zeugmann (1993c)). Therefore, Theorem 7 completely characterizes finite learning. But it does not characterize class comprising dual strong-monotonic inference. The characterization of $CSMON^d - TXT$ remains open. In the following subsection we present further applications of the characterizations obtained.

3.4. Applications

This subsection is devoted to applications of the theorems achieved so far to particular learning problems as well as to further theoretical questions. As outlined in the Introduction, the main goal of this paper consists in answering the question of what a natural learning algorithm should look like. Comparing the IIMs presented in the proofs of Theorem 1 through 7 we see that there is essentially one learning algorithm. Moreover, the difference in the learning power of weak-monotonic, strong-monotonic, and dual strong-monotonic inference is completely characterized by the different properties of the relevant uniformly recursively generable families of finite tell-tale sets. Therefore, the question of whether a particular indexed family may be inferred under a certain monotonicity constraint reduces to the following: Prove or disprove the existence of a tell-tale family that fulfills the relevant properties. Let us exemplify this approach using the pattern languages over some fixed finite alphabet Σ.

First, recall our discussion in the Introduction, where we mentioned that one goal of our study was to determine under what monotonicity constraints the pattern languages are exactly learnable. 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 left to 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)).

Lemma 9. $PAT \in EWMON-TXT$.

Proof. For every pattern $p \in Patc$ we define $T(p)$ to be precisely the set of all strings $s \in L(p)$ that have minimal length. Obviously, $(T(p_j))_{j \in \mathbb{N}}$ is a uniformly recursively generable family of finite and non-empty sets that fulfills Property (1) through (3) of Theorem 2. Hence, $PAT \in EWMON-TXT$. Q.E.D.

Furthermore, a closer look at the IIM defined in the proof of Theorem 2 shows that PAT can be learned by an IIM that is both consistent and conservative.

Next we ask whether we can strengthen the result above. This question is mainly motivated by the following observation. Since we are dealing with learning in the limit,

a potential user of our learning algorithm never knows whether it has already converged. Consequently, she has to use the actual hypothesis as if it were a correct one. Clearly, under such circumstances it is obviously preferable to learn strong-monotonically or monotonically. Our next theorem establishes the equivalence of exact and class preserving strong-monotonic and monotonic learning provided the indexed family contains the maximal element with respect to set inclusion.

Theorem 10. Let $\mathcal{L} = (L_j)_{j \in \mathbb{N}}$ be an indexed family such that $L = \bigcup_{j \in \mathbb{N}} L_j \in \mathcal{L}$. Then $\mathcal{L} \in \lambda SMON{-}TXT$ if and only if $\mathcal{L} \in \lambda MON{-}TXT$ for $\lambda \in \{E, \varepsilon\}$.

Proof. Since the 'if' direction is obvious, it suffices to show that $\mathcal{L} \in \lambda MON{-}TXT$ implies $\mathcal{L} \in \lambda SMON{-}TXT$. Let M be an IIM and let $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ be a hypothesis space such that $\mathcal{L} \in \lambda MON{-}TXT$ w.r.t. $\mathcal{G}$. We show that M strong-monotonically infers $\mathcal{L}$ w.r.t. $\mathcal{G}$. Let $\hat{L} \in \mathcal{L}$ and $\hat{t}$ be a text for $\hat{L}$. Furthermore, let $j_x = M(\hat{t}_x)$ and $j_{x+r} = M(\hat{t}_{x+r})$ be any two consecutive hypotheses output by M when fed $\hat{t}$. Taking into account that $L = \bigcup_{j \in \mathbb{N}} L_j$ we get that $\hat{t}_x$ as well as $\hat{t}_{x+r}$ are initial segments of a text t for L. By the monotonicity constraint we get that $L(G_{j_x}) \cap L \subseteq L(G_{j_{x+r}}) \cap L$. Since $\mathcal{G}$ is a class preserving hypothesis space we may conclude that $L(G_{j_x}) \cap L = L(G_{j_x})$ as well as $L(G_{j_{x+r}}) \cap L = L(G_{j_{x+r}})$. Hence, M learns $\hat{L}$ strong-monotonically. Q.E.D.

The latter theorem easily applies to the pattern languages, because $L(x_0) = \Sigma^+$. Consequently, it suffices to ask whether $PAT \in SMON{-}TXT$. The following lemmata reduce this question to the decidability of $L(p) \subseteq L(q)$ for all $p, q \in Patc$.

Lemma 11. Let $\mathcal{L}$ be an indexed family such that $\mathcal{L} \in SMON{-}TXT$. Then there exists a class preserving hypothesis space $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ such that $L(G_k) \subseteq L(G_j)$ is uniformly decidable for all $k, j \in \mathbb{N}$.

Proof. Let $\mathcal{L} \in SMON{-}TXT$. By Theorem 4 there are a class preserving hypothesis space $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ and a uniformly recursively generable family $(T_j)_{j \in \mathbb{N}}$ of finite and non-empty sets such that Properties (1) through (3) are fulfilled. Let $k, j \in \mathbb{N}$ be arbitrarily fixed. By Property (2) we know that $T_k \subseteq L(G_k)$ and $T_j \subseteq L(G_j)$. Therefore, $L(G_k) \subseteq L(G_j)$ implies $T_k \subseteq L(G_j)$. On the other hand, if $T_k \subseteq L(G_j)$ then, by Property (3), $L(G_k) \subseteq L(G_j)$. Hence, $L(G_k) \subseteq L(G_j)$ if and only if $T_k \subseteq L(G_j)$. Since the sets $(T_j)_{j \in \mathbb{N}}$ are uniformly recursively generable and finite, the lemma follows. Q.E.D.

However, in general the above lemma does not imply the decidability of the inclusion problem for the indexed family $\mathcal{L}$ itself. Nevertheless, there is a special case that applies to the pattern languages.

Lemma 12. Let $\mathcal{L}$ be an indexed family. Then $\mathcal{L} \in SMON{-}TXT \cap FIN{-}INF$ implies the decidability of the inclusion problem for $\mathcal{L}$.

Proof. By Lemma 11 there exists a class preserving hypothesis space $\mathcal{G}$ such that $\mathcal{L} \in SMON{-}TXT$ w.r.t. $\mathcal{G}$, and $L(G_k) \subseteq L(G_j)$ is uniformly decidable for all $k, j \in \mathbb{N}$. Moreover, using the same ideas as in the proof of Corollary 8 it is not hard to show that there is an IIM M that finitely infers $\mathcal{L}$ from informant with respect to the same hypothesis space $\mathcal{G}$. The only difference is that one has to apply the construction outlined in the characterization of $FIN{-}INF$ instead of the one described in Theorem 7 (cf. Lange

and Zeugmann (1994)). Now, let $m, z \in \mathbb{N}$ be arbitrarily fixed. In order to decide whether or not $L_m \subseteq L_z$ do the following. For $x = 0, 1, 2, ...$ compute the initial segments i_x of the lexicographically ordered informant i for L_m of length $x+1$ until the IIM M, when fed i_x, outputs a guess k. Furthermore, for $y = 0, 1, 2, ...$ compute the initial segments i_y of the lexicographically ordered informant i for L_z of length $y+1$ until the IIM M, when fed i_y, outputs a guess j. Since M witnesses $\mathcal{L} \in FIN{-}INF$, we know that $L_m = L(G_k)$ and $L_z = L(G_j)$. Hence, the question whether or not $L_m \subseteq L_z$ is reduced to whether or not $L(G_k) \subseteq L(G_j)$. The latter inclusion is decidable by Lemma 11. Hence, the lemma follows. Q.E.D.

The latter lemma directly applies to PAT. First, $PAT \in FIN-INF$ (Lange and Zeugmann (1992, 1993a)). Next, Jiang et al. (1993) have shown that there is no algorithm deciding $L(p) \subseteq L(q)$ for all $p, q \in Patc$. Hence, we get the following corollary.

Corollary 13. $PAT \notin SMON{-}TXT$.

But still, this is not the whole story. Consider the next theorem.

Theorem 14. $PAT \in CSMON-TXT$.

Proof. By definition $PAT \subseteq DPAT$. Furthermore, there is a class preserving hypothesis space $\mathcal{G}$ such that $DPAT \in SMON{-}TXT$ with respect to $\mathcal{G}$ (cf. Lange and Zeugmann (1992, 1993a)). Thus, $PAT \in CSMON-TXT$ with respect to $\mathcal{G}$. Q.E.D.

Some of the above results give alternative proofs for established results. For example, Corollary 13 and Theorem 14 show that $SMON-TXT \subset CSMON-TXT$. Moreover, taking Lemma 9 into account we additionally get $EWMON{-}TXT \setminus SMON{-}TXT \neq \emptyset$.

Next we outline applications of the characterization theorems that provide additional insight into the problem of what a natural learning algorithm should look like. The general question can be posed as follows: Are there additional, useful properties that learning algorithms can fulfill without restricting their learning capabilities?

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. Another favorable property is order independence. By definition, the output of an IIM may depend on the *range*, the *order*, and the *length* of its actual input. Clearly, the best one hopes to achieve is that its output exclusively depends on the range of its input. IIMs fulfilling this property are said to be ***set-driven*** (cf. Wexler and Culicover (1980)). However, in general set-drivenness is a very restrictive requirement (cf. Osherson et al. (1986), Fulk (1990)). On the other hand, Fulk (1990) proved that any IIM M may be replaced by an IIM M' which is *rearrangement-independent*. An IIM M is said to be ***rearrangement-independent*** iff its output depends only on the *range* and *length* of its input. Unfortunately, Fulk's transformation does not preserve any type of monotonicity. Nevertheless, strong-monotonic inference may always be performed by a rearrangement-independent IIM as the proof of Theorem 4 shows. But this particular IIM is not necessarily semantically finite. It may, however, be replaced by an IIM M' that is *rearrangement-independent* and *semantically finite* and that fulfills the strong-monotonicity constraint. In fact, M' works exactly as M does but it uses a possibly different hypothesis space. By Lemma 11 we know that inclusion for languages

is decidable. Therefore, equality of languages is decidable, too. Hence, we may replace the hypothesis space $\mathcal{G}$ constructed in the proof of Theorem 4 by a hypothesis space $\hat{\mathcal{G}}$ that contains each language exactly ones. Surprisingly enough, the IIM described in the proof of Theorem 1 is semantically finite but not rearrangement-independent. But again, it may be replaced by an IIM M' that is both, semantically finite and rearrangement-independent, and that satisfies the weak-monotonicity constraint. For that purpose one has to redefine the relevant families of uniformly recursively generable tell-tale sets. Moreover, this construction is uniform in the tell-tale families defined in the proof of Theorem 1. We have investigated these questions in detail. The curious reader is referred to Lange and Zeugmann (1993e).

3.5. The Characterization of Monotonic and Dual Monotonic Inference

Next, we characterize $MON-TXT$. As it turns out, characterizing $MON-TXT$ is much more complicated. Intuitively this is due to the following observations: One has to construct a uniformly recursively generable family of finite tell-tales that should contain information concerning both, the corresponding language as well as concerning possible intersections of this language L with those languages L' which may be taken as candidate hypotheses. However, these intersections may yield languages outside the indexed family. Moreover, as long as the output of the IIM M performing the monotonic inference really depends on the *range*, the *order* and the *length* of M's input one has to deal with a *non-recursive* component. The non-recursiveness directly results from the requirement that M has to infer each $L \in \mathcal{L}$ from any text, i.e., one has to find suitable approximations of the uncountable many non-recursive texts. However, assuming that an indexed family can be monotonically learned by a semantically finite and rearrangement-independent IIM (abbr. $\mathcal{L} \in MONR-TXT$) one can characterize its monotonic inferability *mutatis mutandis* as in the case of weak-monotonic or strong-monotonic learning.

Theorem 15. *Let $\mathcal{L}$ be an indexed family. Then $\mathcal{L} \in MONR-TXT$ if and only if there are a space of hypotheses $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ and a recursively generable family $(T_j)_{j \in \mathbb{N}}$ of finite and non-empty sets such that*

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

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

(3) *for all k, $j \in \mathbb{N}$, and for all $L \in \mathcal{L}$, $T_k \subseteq L(G_j) \cap L$ and $T_j \subseteq L$ implies $L(G_k) \cap L \subseteq L(G_j) \cap L$.*

On the other hand, for weak-monotonic, strong-monotonic, and dual strong-monotonic inference we could successfully apply the characterization theorems to prove that semantic finiteness and rearrangement-independence do not restrict their learning power. Now, we have the opposite situation. This led to the conjecture that rearrangement-independence is a severe restriction for exact and class preserving monotonic language learning from positive data. And indeed, this conjecture is true (cf. Lange and Zeugmann (1993e)).

Therefore, in characterizing $MON-TXT$ we have to overcome the difficulties pointed out in a different way.

We now present a characterization of $MON-TXT$ in terms of recursively generable finite sets as above. Additionally, 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 16. *Let $\mathcal{L}$ be an indexed family. Then $\mathcal{L} \in MON$–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) $\mathrm{range}(\mathcal{L}) = L(\mathcal{G})$,

(2) *for all $k \in \mathbb{N}$, $T_k \subseteq L(G_k)$,*

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

(4) *for all $L \in \mathcal{L}$, there is no infinite sequence $(k_j)_{j \in \mathbb{N}}$ such that for all $j \in \mathbb{N}$, $k_j \prec k_{j+1}$ and $\bigcup_{j \in \mathbb{N}} T_{k_j} = L$,*

(5) *for all $L \in \mathcal{L}$ and all k, $j \in \mathbb{N}$, if $k \prec j$, $T_j \subseteq L$, then $L(G_k) \cap L \subseteq L(G_j) \cap L$.*

Proof. Necessity: Let M be an IIM inferring $\mathcal{L}$ without loss of generality monotonically and consistently with respect to some space $\hat{\mathcal{G}} = (\hat{G}_j)_{j \in \mathbb{N}}$ of hypotheses (cf. Lange and Zeugmann (1992, 1993a)). First, we construct a space of hypotheses $\tilde{\mathcal{G}} = (\tilde{G}_j)_{j \in \mathbb{N}}$ 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 $\tilde{G}_{c(n,x)} = \hat{G}_n$. The family $(\tilde{T}_{c(n,x)})_{n,x \in \mathbb{N}}$ is defined as follows.

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

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

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

Let t be the canonical text of L. Since M learns L, there exists $n, y \in \mathbb{N}$ 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, $\tilde{T}_{c(n,x)} \neq \emptyset$, and $L(\tilde{G}_{c(n,x)}) = L$. 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 $k, j \in \mathbb{N}$, and let m, n, x, y be the uniquely determined numbers such that $f(k) = c(m,y)$ and $f(j) = c(n,x)$. Then we define $k \prec j$ 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 (5) 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 $k \in \mathbb{N}$ be any index such that $T_k \subseteq L$ and $L(G_k) \neq L$. We have to show that there is a j with $k \prec j$ and $A \subseteq T_j \subseteq L(G_j) = L$. In accordance with our construction we have $T_k = \tilde{T}_{f(k)}$ and $G_k = \tilde{G}_{f(k)}$. Let m, y be the uniquely determined numbers with $f(k) = c(m,y)$. Then we know that $M(\sigma_y) = m$ and $L \neq L(\hat{G}_m)$, since $L(\hat{G}_m) = L(\tilde{G}_{c(m,y)}) = L(G_k) \neq L$. Moreover, $T_k = \sigma_y^+ \subseteq L$. Hence, σ_y is an initial segment of a text for L. Since $L \neq L(G_k)$, 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 $\emptyset \neq \tilde{T}_{c(n,x)} \subseteq L(\tilde{G}_{c(n,x)}) = L$. Thus, there is a number j such that $f(j) = c(n,x)$, and for every j with $f(j) = c(n,x)$ we get $\sigma_y \sqsubset \sigma_x$ and $m \neq n$. Therefore, $k \prec j$, and (3) is proved.

We proceed with the demonstration of (4). Looking at the definition of the relation $\prec$ we immediately realize that $k \prec j$ implies $T_k \subseteq T_j$. Suppose there is an infinite sequence $(k_j)_{j \in \mathbb{N}}$ such that $k_j \prec k_{j+1}$ and $\bigcup_{j \in \mathbb{N}} T_{k_j} = L$. Since $T_{k_j} \subseteq T_{k_{j+1}}$, we get in the limit a text t of L on which M changes its mind infinitely often, a contradiction. Hence, (4) is proved.

Finally, we demonstrate (5). Let $L \in \mathcal{L}$, and let $k, j \in \mathbb{N}$ such that $k \prec j$ and $T_j \subseteq L$. Furthermore, let $m, n, x, y \in \mathbb{N}$ be the uniquely determined numbers such that $f(k) = c(m,y)$ and $f(j) = c(n,x)$. Recalling the definition of the relation $\prec$ we get $\sigma_y \sqsubset \sigma_x$ as well as $m \neq n$. Moreover, $M(\sigma_y) = m$ and $M(\sigma_x) = n$ by the definition of the tell-tale family. Since $T_j = \sigma_x^+$ and $T_j \subseteq L$, we see that σ_x is an initial segment of some text t for L on which M, when successively fed it, outputs m and in some subsequent step n. Taking into account that M works monotonically, we obtain $L(\hat{G}_m) \cap L \subseteq L(\hat{G}_n) \cap L$. Finally, in accordance with our construction we know that $L(G_k) = L(\tilde{G}_{c(m,y)}) = L(\hat{G}_m)$ as well as $L(G_j) = L(\tilde{G}_{c(n,x)}) = L(\hat{G}_n)$. Hence, (5) follows.

Sufficiency: Again, it suffices to describe an IIM M that infers $\mathcal{L}$ monotonically 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 $k \leq x$ such that $T_k \subseteq t_x^+$. In case it is found, set $y_k = x$, output k and request the next input. Otherwise, output nothing and request the next input.

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

Claim 1. If M converges, say to k, then $L = L(G_k)$.

First observe that $T_k \subseteq L$, since otherwise k cannot be any of M's guesses. Suppose $L(G_k) \neq L$. By Property (3), there exists a j such that $k \prec j$ and $t_{y_k} \subseteq T_j \subseteq L(G_j) = L$. Hence, there should be an $x \in \mathbb{N}$ such that $t_{y_k}^+ \subseteq T_j \subseteq t_x^+$. Consequently, M on input t_x has to change its mind from k to j, a contradiction.

Claim 2. M works monotonically.

This is an immediate consequence of Property (5) and the definition of M.

Claim 3. M converges on t.

In accordance with (1) and (2), one can show analogous to the proof of Theorem 1 that M outputs at least once a hypothesis, say k. Moreover, as long as M does not find a j such that $k \prec j$, and $t_{y_k}^+ \subseteq T_j \subseteq t_x^+$, this guess is repeated. Hence, as long as M finds only finitely many j's in (B), it converges. Consequently, if M does not converge, it finds an infinite sequence $(k_j)_{j \in \mathbb{N}}$ such that $k_j \prec k_{j+1}$ for all $j \in \mathbb{N}$. Furthermore, every mind change implies an update of the value of variable y_{k_j}. Thus, for all $z \in \mathbb{N}$ there exist $y_{k_j}, x \in \mathbb{N}$ with $t_z^+ \subseteq t_{y_{k_j}}^+ \subseteq T_{k_j} \subseteq t_x^+$. Therefore, we immediately obtain $\bigcup_{j \in \mathbb{N}} T_{k_j} = L$, a contradiction to Property (4). This proves the claim, and hence the sufficiency is shown. Thus, the theorem is proved. Q.E.D.

Next we characterize dual monotonic language learning as well as $MON^{\&}\text{-}TXT$. As it turned out, the proof technique developed above is powerful enough to characterize $MON^{d}\text{-}TXT$ as well.

Theorem 17. *Let $\mathcal{L}$ be an indexed family. Then $\mathcal{L} \in MON^{d}\text{-}TXT$ if and only if there are a space of hypotheses $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$, 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) $\mathrm{range}(\mathcal{L}) = L(\mathcal{G})$,

(2) *for all* $k \in \mathbb{N}$, $T_k \subseteq L(G_k)$,

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

(4) *for all* $L \in \mathcal{L}$, *there is no infinite sequence* $(k_j)_{j \in \mathbb{N}}$ *such that for all* $j \in \mathbb{N}$, $k_j \prec k_{j+1}$ *and* $\bigcup_j \hat{T}_{k_j} = L$,

(5) *for all* $L \in \mathcal{L}$ *and all* k, $j \in \mathbb{N}$, *if* $k \prec j$, $T_j \subseteq L$, *then* $co-L(G_k) \cap co-L \subseteq co-L(G_j) \cap co-L$.

Proof. For a proof the reader is referred to Zeugmann *et al.* (1992). Q.E.D.

Note that the IIM defined in the proof of Theorem 16 uses a new technique to detect that its actual hypothesis may be incorrect. Clearly, no IIM can prove that its actual guess is really correct, unless 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 of Theorem 16 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 for detecting 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)).

We next characterize $MON^{\&}-TXT$.

Theorem 18. *Let* $\mathcal{L}$ *be an indexed family. Then* $\mathcal{L} \in MON^{\&}-TXT$ *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$(\mathcal{L}) = L(\mathcal{G})$,

(2) *for all* $k \in \mathbb{N}$, $T_k \subseteq L(G_k)$,

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

(4) *for all* $L \in \mathcal{L}$, *there is no infinite sequence* $(k_j)_{j \in \mathbb{N}}$ *such that for all* $j \in \mathbb{N}$, $k_j \prec k_{j+1}$ *and* $\bigcup_j T_{k_j} = L$,

(5) *For all* $L \in \mathcal{L}$ *and all* k, $j \in \mathbb{N}$, *if* $k \prec j$, $T_j \subseteq L$, *then*

(i) $L(G_j) \setminus L(G_k) \subseteq L$, *and*
(ii) $(L(G_k) \setminus L(G_j)) \cap L = \emptyset$.

Proof. Necessity: Properties (1) through (4) can be proved as in Theorem 16. Hence, it suffices to show that if, for any $k, j \in \mathbb{N}$ and $L \in \mathcal{L}$, $k \prec j$, $T_j \subseteq L$, then

(i) $L(G_j) \setminus L(G_k) \subseteq L$, and

(ii) $(L(G_k) \setminus L(G_j)) \cap L = \emptyset$.

By construction, $k \prec j$ implies that there are $m, n, x, y \in \mathbb{N}$ such that $M(\sigma_y) = m$, $M(\sigma_x) = n$, $\sigma_y \sqsubset \sigma_x$ as well as $L(\tilde{G}_{c(m,y)}) = L(G_k)$ and $L(\tilde{G}_{c(n,x)}) = L(G_j)$. Hence, when successively fed σ_x the IIM M eventually outputs m and in some subsequent step n. Since $\sigma_x^+ \subseteq T_j \subseteq L$, it follows that σ_x constitutes an initial segment of a text t for L. We claim that $L(G_j) \setminus L(G_k) \subseteq L$. Suppose the converse, i.e., there is a string s such that $s \in L(G_j) \setminus L(G_k)$ and $s \notin L$. Hence, $s \in co{-}L \cap co{-}L(G_k)$ but $s \notin co{-}L \cap co{-}L(G_j)$. Therefore, $co{-}L(G_k) \cap co{-}L \not\subseteq co{-}L(G_j) \cap co{-}L$ and consequently, M does not work dual monotonically. Thus, (i) is proved. We proceed to show (ii). Suppose there is a string $s \in (L(G_k) \setminus L(G_j)) \cap L$. Then $s \in L(G_k) \cap L$ and $s \notin L(G_j)$. Therefore, $s \notin L(G_j) \cap L$. However, the latter statement contradicts the monotonicity requirement $L(G_k) \cap L \subseteq L(G_j) \cap L$. Hence, (ii) is proved.

Sufficiency: The sufficiency can be shown *mutatis mutandis* as in the proof of Theorem 16. Therefore, it is omitted here. Q.E.D.

4. Conclusions and Open Problems

Different notions of monotonicity and of dual monotonicity have been defined and almost all the resulting models of inference from positive data have been characterized in terms of finitely generable tell-tale sets. In particular, the characterizations of weak-monotonic learning comprehensively solved the longstanding open problem of how to characterize inference algorithms that avoid overgeneralization. Furthermore, applying the established characterization for strong-monotonic learning as well as the recently obtained result by Jiang et al. (1993), we completely solved the open problem of Jantke (1991b) of whether or not the pattern languages are strong-monotonically inferable from text.

Moreover, we obtained a unifying approach to monotonic language learning by describing general algorithms that perform any monotonic inference task. As we have seen, *essentially* one easily programmable learning algorithm may be used. Hence, we additionally obtain an answer to the question of what is more appropriate in a specific situation, learning by generalization or learning by specialization. All the differences between these two global strategies can be expressed in terms of properties the hypothesis space and the finite tell-tale sets must have. Thus, deciding what learning strategy one should choose to solve a particular learning problem reduces to elaborating characteristic features of the target indexed family that may be incorporated into suitable descriptions of the target objects and that can be described by finite tell-tale sets. As demonstrated in Subsection 3.4, it is the interplay between the properties of the hypothesis space and the relevant tell-tale sets that makes or does not make a learning problem solvable. We present one more example to illustrate this insight. As we have seen, the decidability of set inclusion of languages is necessary for $SMON{-}TXT$ identification. However, it is not sufficient. In Lange and Zeugmann (1992, 1993a), we have shown that there is an

indexed family for which set inclusion is uniformly decidable but is not *monotonically* inferable, even from *informant*. Next, the established characterization theorems may help to answer the question whether or not natural learning algorithms do almost always exist. For example, one may investigate the problem whether the combination of certain monotonicity constraints with additional desirable requirements, e.g., set-drivenness, rearrangement-independence, or semantic finiteness, does seriously affect the capabilities of learning algorithms. We have successfully attacked this problem by applying the characterizations presented in the present paper. For more information, the reader is referred to Lange and Zeugmann (1993e).

It has been suggested that our tell-tale subset characterizations may have connections with the concept of key arrays in recursive function theory (cf. Rice (1956)). In fact, the two concepts are quite different. For one, in the case of completely r.e. sets, *all* indices of the language are included, not *some*. Furthermore, the key arrays are also part of the completely r.e. sets, while the tell-tale subsets (for infinite languages, in particular) are definitely not part of the family. In the case of tell-tale subsets, we are only concerned that there be no intervening language (between the tell-tale subset and the language whose tell-tale it is) in the family. In case of key arrays, all the r.e. supersets of the key arrays must be in the completely r.e. set.

Finally, we point out another interesting aspect of Angluin's (1980b) as well as of our characterizations. Freivalds *et al.* (1993) introduced inference from good examples, i.e., instead of successively inputting the whole graph of a function now an IIM obtains only a finite set of pairs (argument,value)—the so-called "good examples." Then, it finitely infers a function iff it outputs a single correct hypothesis. Surprisingly, finite inference of recursive functions from good examples is *more powerful* than identification in the limit. The same approach may be undertaken in language learning (cf. Lange and Wiehagen (1991)). Obviously the most exciting case is to consider learning from good examples exclusively chosen from the language to be learned. We conjecture that the finite tell-tale sets defined in accordance with our characterizations may also serve as sets of good examples.

Furthermore, as our results show, all types of monotonic language learning have special features distinguishing them from monotonic inference of recursive functions. Therefore, it would be very interesting to study monotonic language learning in the general case, i.e., not restricted to indexed families.

Acknowledgement

The first two authors gratefully acknowledge many enlightening discussions with Rolf Wiehagen concerning the characterization of learning algorithms. We heartily thank the anonymous referees, whose comments improved the paper immensely.

5. References

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, *In-*

formation and Control **45**, 117 - 135.

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.

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.

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

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

BLUM, L., AND BLUM, M. (1975), Toward a mathematical theory of inductive inference, *Information and Control* **28**, 122 - 155.

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," (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.

FREIVALDS, R., KINBER, E.B., AND WIEHAGEN, R. (1993), On the power of inductive inference from good examples, *Theoretical Computer Science* **110**, 131 - 144.

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, Morgan Kaufmann Publishers Inc., San Mateo.

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, ACM Press, New York.

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

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, Springer-Verlag, Berlin.

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.

JIANG, T., SALOMAA, A., SALOMAA, K. AND YU, S. (1993), Inclusion is undecidable for pattern languages, *in* "Proceedings 20th International Colloquium on Automata, Languages and Programming, July 1993, Lund," (A. Lingas, R. Karlsson, S. Carlsson, Eds.), Lecture Notes in Computer Science Vol. 700, pp. 301 - 312, Springer-Verlag, Berlin.

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, also *Theoretical Computer Science* **141** No. 1-2, 151 - 162, (1995).

KEARNS, M., AND PITT, L. (1989), A polynomial-time algorithm for learning k–variable pattern languages from examples, *in* "Proceedings 2nd Annual Workshop on Computational Learning Theory," (R. Rivest, D. Haussler and M.K. Warmuth, Eds.), pp. 57 -71, Morgan Kaufmann Publishers Inc., San Mateo.

KODRATOFF, Y., AND MICHALSKI, R.S. (1990), "Machine Learning, An Artificial Intelligence Approach," Vol. 3, Morgan Kaufmann Publishers Inc., San Mateo.

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), On the power of monotonic language learning, GOSLER–Report 05/92, February 1992, Fachbereich Mathematik und Informatik, TH Leipzig.

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

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), Characterization of finite identification, *in* "Proceedings 3rd International Workshop on Analogical and Inductive Inference, October 1992, Dagstuhl Castle," (K.P. Jantke, Ed.), Lecture Notes in Artificial Intelligence Vol. 642, pp. 260 - 267, Springer-Verlag, Berlin.

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.

RICE, H.G. (1956), On completely recursively enumerable classes and their key arrays, *The Journal of Symbolic Logic* **21**, 304 - 308.

RIVEST, R., HAUSSLER, D., AND WARMUTH, M.K. (Eds.) (1989), Proceedings of the 2nd Annual Workshop on Computational Learning Theory, Morgan Kaufmann Publishers Inc., San Mateo.

SHINOHARA, T. (1990), Inductive inference from positive data is powerful, *in* "Proceedings 3rd Annual Workshop on Computational Learning Theory, Rochester, August 1990, (M.A. Fulk and J. Case, Eds.), pp. 97 - 110. Morgan Kaufmann Publishers Inc., San Mateo.

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., AND CULICOVER, P. (1980), "Formal Principles of Language Acquisition," MIT-Press, Cambridge, Massachusetts.

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

WIEHAGEN, R. (1991), A thesis in inductive inference, *in* "Proceedings 1st International Workshop on Nonmonotonic and Inductive Logic, December 1990, Karlsruhe," (J. Dix, K.P. Jantke and P.H. Schmitt, Eds.), Lecture Notes in Artificial Intelligence Vol. 543, pp. 184 - 207, Springer-Verlag, Berlin.

WIEHAGEN, R., AND LIEPE, W. (1976), Charakteristische Eigenschaften von erkennbaren Klassen rekursiver Funktionen, *Journal of Information Processing and Cybernetics (EIK)* **12**, 421 - 438.

ZEUGMANN, T. (1983), A–posteriori characterizations in inductive inference of recursive functions, *Journal of Information Processing and Cybernetics (EIK)* **19**, 559 - 594.

ZEUGMANN, T. (1988), On the power of recursive optimizers, *Theoretical Computer Science* **62**, 289 - 310.

ZEUGMANN, T., LANGE, S., AND KAPUR, S. 1992, Characterizations of Class Preserving Monotonic and Dual Monotonic Language Learning, Technical Report IRCS 92 - 24, Institute for Research in Cognitive Science, University of Pennsylvania, Philadelphia.