Characterization of Language Learning from Informant under Various Monotonicity Constraints

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

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

Abstract

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

The three versions of dual monotonicity describe the concept that the inference device has to produce exclusively specializations that fit better and better to the target language. We characterize strong–monotonic, monotonic, weak–monotonic, dual strong–monotonic, dual monotonic and dual weak–monotonic as well as finite language learning from positive and negative data in terms of recursively generable finite sets. Thereby, we elaborate a unifying approach to monotonic language learning by showing that there is exactly one learning algorithm which can perform any monotonic inference task.

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. Since the seminal papers of Solomonoff (1964) and Gold (1967), problems in inductive inference have additionally found a lot of attention from computer scientists. The theory they have developed within the last decades is usually referred to as *computational or algorithmic learning theory.* The state of the art of this theory is excellently surveyed in Angluin and Smith (1983, 1987).

Within the present paper we deal with identification of formal languages. Formal language learning may be considered as inductive inference of partial recursive functions. Nevertheless, some of the results are surprisingly in that they remarkably differ from solutions for analogous problems in the setting of inductive inference of recursive functions (cf. e.g. Osherson, Stob and Weinstein (1986), Case (1988), Fulk (1990)).

The general situation investigated in language learning can be described as follows: Given more and more eventually incomplete information concerning the language to be learnt, the inference device has to produce, from time to time, a hypothesis about the phenomenon to be inferred. The information given may contain only *positive examples*, i.e., exactly all the strings contained in the language to be recognized, as well as both *positive and negative examples*, i.e., arbitrary strings over the underlying alphabet which are classified with respect to their containment to the unknown language. The sequence of hypotheses has to converge to a hypothesis which correctly describes the language to be learnt. In the present paper, we mainly study language learning from positive and negative examples.

Monotonicity requirements have been introduced by Jantke (1991A, 1991B) and Wiehagen (1991) in the setting of inductive inference of recursive functions. We have adopted their definitions to the inference of formal languages (cf. Lange and Zeugmann (1992A, 1992B, 1993)). Subsequently Kapur (1992) introduced the dual versions of monotonic language learning. The main underlying question can be posed as follows: Would it be possible to infer the unknown language in a way such that the inference device *only* outputs better and better generalizations and specializations, respectively?

The strongest interpretation of this requirement means that we are forced to produce an augmenting (descending) chain of languages, i.e., $L_i \subseteq L_j$ ($L_i \supseteq L_j$) iff L_j is guessed later than L_i (cf. Definition 3 and 5, part (A)).

Wiehagen (1991) proposed to interpret "better" with respect to the language L having to be identified, i.e., now we require $L_i \cap L \subseteq L_j \cap L$ iff L_j appears later in the sequence of guesses than L_i does (cf. Definition 3 (B)). That means, a new hypothesis is never allowed to destroy something what 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 speaking, dual monotonicity describes the following requirement. If the learner outputs at any stage a hypothesis correctly excluding a string s from the language to be learnt, then any subsequent guess has to behave thus (cf. Definition 5 (B)). The third version of monotonicity, which we call weak–monotonicity and dual weak–monotonicity, respectively, is derived from non–monotonic logics and adopts the concept of cumulativity and of its dual analogue, respectively. Hence, we only require $L_i \subseteq L_j$ ($L_i \supseteq L_j$) as long as there are no data fed to the inference device after having produced L_i that contradict L_i (cf. Definition 3 and 5, part (C)).

In all what follows, we restrict ourselves to deal exclusively with the learnability of indexed families of non–empty uniformly recursive languages (cf. Angluin(1980)). This case is of special interest with respect to potential applications. The first problem arising naturally is to relate all types of monotonic and of dual monotonic language learning one to the other as well as to previously studied modes of inference. Concerning monotonic language learning this question has been almost completely answered by Lange and Zeugmann (1992A, 1993). Dual monotonic inference of languages from positive data has been introduced in Kapur (1992) and intensively studied in Lange, Zeugmann and Kapur (1992). In the sequel we deal with the different modes of monotonic and of dual monotonic language learning from positive and negative data. As it turns out, weak–monotonically and dual weak–monotonically working learning devices from positive and negative data are exactly as powerful as *conservatively* working ones, as we shall show. A learning algorithm is said to be *conservative* iff it only performs justified mind changes. That means, the learner may change its guess only in case if the former hypothesis "provably misclassifies" some word with respect to the data seen so far. Considering learning from positive and negative examples in the setting of indexed families it is not hard to prove that conservativeness does not restrict the inference capabilities. Surprisingly enough, in the setting of learning recursive functions the situation is totally different (cf. Freivalds, Kinber and Wiehagen (1992)). Another interesting problem consists in characterizing monotonic language learning. In general, characterizations play an important role in inductive inference (cf. e.g. Wiehagen (1977, 1991), Angluin (1980), Freivalds, Kinber and Wiehagen (1992)). On the one hand, they allow to state precisely what kind of requirements a class of target objects has to fulfil in order to be learnable from eventually incomplete data. On the other hand, they lead to deeper insights into the problem how algorithms performing the desired learning task may be designed. Angluin (1980) proved a characterization theorem for language learning from positive data that turned out to be very useful in applications. In Lange and Zeugmann (1992B), we adopt the underlying idea for characterizing all types of monotonic language learning from positive data in terms of recursively generable finite sets.

Because of the strong relation between inductive inference of recursive functions and language learning from informant, one may conjecture that the characterizations for monotonic learning of recursive functions (cf. Wiehagen (1991), Freivalds, Kinber and Wiehagen (1992)) do easily apply to monotonic language learning. However, monotonicity requirements in inductive inference of recursive functions are defined with respect to the graph of the hypothesized functions. This makes really a difference as the following example demonstrates. Let $L \subseteq \Sigma^*$ be any arbitrarily fixed *infinite* context-sensitive language. By $\mathcal{L}_{fin}$ we denote the set of all finite languages over Σ. Then we set $\mathcal{L}_{finvar} = \{L \cup L_{fin} \mid L_{fin} \in \mathcal{L}_{fin}\}$. In our setting, $\mathcal{L}_{finvar}$ is strong–monotonically learnable, even on text (cf. Lange and Zeugmann (1992A)). If one uses the same concept of strong–monotonicity as in Freivalds, Kinber and Wiehagen (1992), one immediately obtains from Jantke (1991A) that, even from informant, $\mathcal{L}_{finvar}$ cannot be learnt strong–monotonically. This is caused by the following facts. First, any IIM M that eventually identifies $\mathcal{L}_{finvar}$ strong–monotonically with respect to the graphs of their characteristic functions has to output sometime a program of a recursive function. Next, the first program of a recursive function has to be a correct one. Finally, it is not hard to prove that no IIM M can satisfy the latter requirement.

In order to develop a unifying approach to all types of monotonic and of dual monotonic language learning, we present characterizations of monotonic as well as of dual monotonic language learning from informant in terms of recursively generable finite sets. In doing so, we will show that there is exactly one learning algorithm that may perform each of the desired inference tasks from informant. Moreover, it turns out that a conceptually very close algorithm may be also used for monotonic language learning from positive data (cf. Lange and Zeugmann (1992B)).

2. Preliminaries

By $\mathbb{N} = \{1, 2, 3, ...\}$ we denote the set of all natural numbers. In the sequel we assume familiarity with formal language theory (cf. e.g. Bucher and Maurer (1984)). 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. Let $i = (s_1, b_1), (s_2, b_2), ...$ be an infinite sequence of elements of $\Sigma^* \times \{+, -\}$ such that $range(i) = \{s_k \mid k \in \mathbb{N}\} = \Sigma^*$, $i^+ = \{s_k \mid (s_k, b_k) = (s_k, +), k \in \mathbb{N}\} = L$ and $i^- = \{s_k \mid (s_k, b_k) = (s_k, -), k \in \mathbb{N}\} = co - L$. Then we refer to i as an *informant*. If L is classified via an informant then we also say that L is represented by *positive and negative data*. Moreover, let i be an informant and let x be a number. Then i_x denotes the initial segment of i of length x, e.g., $i_3 = (s_1, b_1), (s_2, b_2), (s_3, b_3)$. Let i be an informant and let $x \in \mathbb{N}$. By i_x^+ and i_x^- we denote the sets $range^+(i_x) := \{s_k \mid (s_k, +) \in i, k \leq x\}$ and $range^-(i_x) := \{s_k \mid (s_k, -) \in i, k \leq x\}$, respectively. Finally, we write $i_x \sqsubseteq i_y$, if i_x is a prefix of i_y.

Following Angluin (1980) we restrict ourselves to deal exclusively with indexed families of recursive languages defined as follows:

A sequence $L_1, L_2, L_3, ...$ is said to be an *indexed family* $\mathcal{L}$ of 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

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

As an example we consider the set $\mathcal{L}$ of all context–sensitive languages over Σ. Then $\mathcal{L}$ may be regarded as an indexed family of recursive languages (cf. Bucher and Maurer (1984)). In the sequel we often denote an indexed family and its range by the same symbol $\mathcal{L}$. What is meant will be clear from the context.

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

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 in dependence 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 and dual monotonicity with respect to the language to be learnt, i.e., to require containment of languages as described in the introduction. The latter approach increases considerably the power of monotonic and of dual monotonic language learning (cf. e.g. the example presented in the introduction). Furthermore, since we exclusively deal with indexed families $\mathcal{L} = (L_j)_{j\in\mathbb{N}}$ of recursive languages we almost always take as space of hypotheses an enumerable family of grammars $G_1, G_2, G_3, ...$ over the terminal alphabet Σ satisfying $\mathcal{L} = \{L(G_j) \mid j \in \mathbb{N}\}$. Moreover, we require that membership in $L(G_j)$ is uniformly decidable for all $j \in \mathbb{N}$ and all strings $s \in \Sigma^*$. As it turns out, it is sometimes very important to choose the space of hypotheses appropriately in order to achieve the desired learning goal. Then the IIM outputs numbers j which we interpret as G_j.

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

Definition 1. (Gold, 1967) *Let $\mathcal{L}$ be an indexed family of languages, $L \in \mathcal{L}$, and let $(G_j)_{j\in\mathbb{N}}$ be a space of hypotheses. An IIM M $LIM-INF$–identifies L on an informant i iff it almost always outputs a hypothesis and the sequence $(M(i_x))_{x\in\mathbb{N}}$ converges in the limit to a number j such that $L = L(G_j)$.*
Moreover, M $LIM-INF$–identifies L, iff M $LIM-INF$–identifies L on every informant for L. We set:
$LIM-INF(M) = \{L \in \mathcal{L} \mid M\ LIM-INF-identifies\ L\}$.
Finally, let $LIM-INF$ denote the collection of all indexed families $\mathcal{L}$ of recursive languages for which there is an IIM M such that $\mathcal{L} \subseteq LIM-INF(M)$.

Definition 1 could be easily generalized to arbitrary families of recursively enumerable languages (cf. Osherson et al. (1986)). Nevertheless, we exclusively consider the restricted case defined above, since our motivating examples are all indexed families of recursive languages. Note that, in general, it is not decidable whether or not M has already inferred L. Within the next definition, we consider the special case that it has to be decidable whether or not an IIM has successfully finished the learning task.

Definition 2. (Trakhtenbrot and Barzdin, 1970) *Let $\mathcal{L}$ be an indexed family of languages, $L \in \mathcal{L}$, and let $(G_j)_{j\in\mathbb{N}}$ be a space of hypotheses. An IIM M $FIN-INF$–identifies L on an informant i iff it outputs only a single and correct hypothesis j, i.e., $L = L(G_j)$, and stops.*
Moreover, M $FIN-INF$–identifies L, iff M $FIN-INF$–identifies L on every informant

for L. We set: $FIN-INF(M) = \{L \in \mathcal{L} \mid M\ FIN-INF-\text{identifies } L\}$.

The resulting identification type is denoted by $FIN-INF$. Next, we formally define strong–monotonic, monotonic and weak–monotonic inference.

Definition 3. (Jantke, 1991A, Wiehagen 1991) *An IIM M is said to identify a language L from informant*

(A) strong–monotonically

(B) monotonically

(C) weak–monotonically

iff

M $LIM-INF$–identifies L and for any informant i of L as well as for any two consecutive hypotheses j_x, j_{x+k} which M has produced when fed i_x and i_{x+k}, for some $k \geq 1, k \in \mathbb{N}$, 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 $i^+_{x+k} \subseteq L(G_{j_x})$ *and* $i^-_{x+k} \subseteq co-L(G_{j_x})$, *then* $L(G_{j_x}) \subseteq L(G_{j_{x+k}})$.

We denote by $SMON-INF$, $MON-INF$, and $WMON-INF$ the collection of all thoses sets $\mathcal{L}$ of indexed families of languages for which there is an IIM inferring it strong–monotonically, monotonically, and weak–monotonically from informant, respectively.

We continue in defining *conservatively* working IIMs. Intuitively speaking, a conservatively working IIM performs *exclusively justified* mind changes. Note that $WMON-INF$ = *CONSERVATIVE–INF* (cf.Lange and Zeugmann (1992A, 1993)).

Definition 4. (Angluin, 1980A)
An IIM M CONSERVATIVE–INF–identifies L from informant i, iff for every informant i the following conditions are satisfied:

(1) $L \in LIM - INF(M)$

(2) If M *on input* 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 either to contain some string* $s \in i^+_{x+k}$ *or it generates some string* $s \in i^-_{x+k}$.

CONSERVATIVE–INF(M) as well as the collection of sets *CONSERVATIVE–INF* are defined in an analogous manner as above.

Finally in this section, we define the corresponding modes of dual monotonic language learning.

Definition 5. (Kapur, 1992) *An IIM M is said to identify a language L from informant*

(A) dual strong–monotonically

(B) dual monotonically

(C) dual weak–monotonically

iff

M $LIM - INF$*–identifies* L *and for any informant* i *of* L *as well as for any two consecutive hypotheses* j_x, j_{x+k} *which* M *has produced when fed* 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 $i^+_{x+k} \subseteq L(G_{j_x})$ *and* $i^-_{x+k} \subseteq co - L(G_{j_x})$, *then* $co - L(G_{j_x}) \subseteq co - L(G_{j_{x+k}})$.

We denote by $SMON^d - INF$, $MON^d - INF$, and $WMON^d - INF$ the family of all thoses sets $\mathcal{L}$ of indexed families of languages for which there is an IIM inferring it dual strong–monotonically, dual monotonically, and dual weak–monotonically from informant, respectively.

3. Monotonic and Dual Monotonic Inference

The aim of the present chapter is to relate the different types of monotonic and of dual monotonic language learning one to the other. Some of the results originate from Lange and Zeugmann (1993).

The following proposition is obvious.

Proposition 1.

(1) $FIN-INF \subseteq SMON-INF \subseteq MON-INF \subseteq WMON-INF \subseteq LIM-INF$

(2) $FIN-INF \subseteq SMON^d-INF \subseteq MON^d-INF \subseteq WMON^d-INF \subseteq LIM-INF$

Our first theorem actually shows what monotonic and dual monotonic language learning from informant have in common and where the differences are.

Theorem 1.

(1) $WMON-INF = WMON^d-INF = LIM-INF$

(2) $MON-INF \# MON^d-INF$

(3) $SMON-INF \# SMON^d-INF$

Proof. For assertion (1) one has simply to recognize that any indexed family $\mathcal{L}$ can be identified from informant by an IIM that works by the *identification by enumeration principle* (cf. Gold (1967)). This IIM performs only justified mind changes. Hence, it works weak–monotonically as well as dual weak–monotonically. The remaining part is shown via the following claims.

Claim A. $MON-INF \setminus MON^d-INF \neq \emptyset$

We set $L_1 = \{a\}^*$, $L_k = \{a^n \mid 1 \leq n \leq k\} \cup \{b^n \mid k < n\}$ for $k \geq 2$, and $L_{k,m} = \{a^n \mid 1 \leq n \leq k\} \cup \{b^n \mid k < n \leq m\} \cup \{c^m\}$ for $m > k \geq 2$. Assume $\mathcal{L}$ to be any appropriate enumeration of all these languages. First, we show that $\mathcal{L} \in MON-INF$. The wanted IIM M which monotonically identifies $\mathcal{L}$ works as follows: As long as some $(a^n, -)$ does not appear in the informant, the machine outputs a grammar for L_1. In case it does, M performs a mind change, when both $(a^k,+)$ and $(b^{k+1},+)$ have been seen. Then, M outputs a grammar for L_k as long as no pair $(c^m,+)$ is presented. If such a pair does appear, M changes its mind and outputs a grammar for $L_{k,m}$. In any subsequent step, M repeats this hypothesis . Obviously, M monotonically identifies $\mathcal{L}$.

It remains to show that $\mathcal{L} \notin MON^d - INF$. Suppose the converse, i.e., there is an IIM M which dual monotonically infers $\mathcal{L}$. Let i be any informant for $\{a\}^*$. Hence, there must be an x such that $j_x = M(i_x)$ and $L(G_{j_x}) = \{a\}^*$. Let $k = max\{|s| \mid s \in i_x^+ \cup i_x^-\}$. Then, we consider any informant $\tilde{i}$ for L_k with $i_x \sqsubseteq \tilde{i}$. Since $L_k \in \mathcal{L}$, there has to be a y such that $j_y = M(\tilde{i}_y)$ and $L(G_{j_y}) = L_k$. Let $m = max\{|s| \mid s \in \tilde{i}_y^+ \cup \tilde{i}_y^-\}$. Obviously, $\tilde{i}_y$ is an initial segment of an informant i_{fool} for $L_{k,m}$. Thus, M either does not work dual monotonically on i_{fool} or it fails to infer $L_{k,m}$. If M produces sometime the hypothesis j_x and afterwards j_y when processing i_{fool}, then $b^{m+1} \in co - L_1 \cap co - L_{k,m}$, but $b^{m+1} \notin co - L_k \cap co - L_{k,m}$ which violates the dual monotonicity requirement.

Claim B. $MON^d - INF \setminus MON - INF \neq \emptyset$

We define an indexed family over $\{a, b\}$ as follows: We set $L_1 = \{a\}^*$, $L_k = \{a^n \mid 1 \leq n \leq k\} \cup \{b^{k+1}\}$ with $k \geq 2$, and $L_{k,m} = L_k \cup \{a^m\}$ with $m > k \geq 2$. Assume $\mathcal{L}$ to be any appropriate enumeration of all these languages. An IIM M which dual monotonically identifies $\mathcal{L}$ may work as follows: Let $L \in \mathcal{L}$ and let i be any informant for L. As long as no $(a^n,-)$ does appear in the informant i, the machine outputs a grammar for L_1. If a pair $(a^n,-)$ is presented, M performs a mind change when both, $(a^k,+)$ and $(b^{k+1},+)$, have been seen. In this case, M outputs a grammar for L_k as long as no pair $(a^m,+)$ with $m > k$ will be presented. If such a pair does appear, M changes its mind to and outputs a grammar for $L_{k,m}$. This hypothesis is then repeated in any subsequent step.

Obviously, M identifies $\mathcal{L}$. It remains to show that M works dual monotonically. By construction, M performs at most two mind changes, i.e., it eventually outputs j_x for L_1, j_{x+z} for L_k and finally j_{x+z+y} for $L_{k,m}$. Then, we have $co - L_1 \cap co - L_{k,m} \subseteq co - L_k \cap co - L_{k,m} \subseteq co - L_{k,m}$, since $co - L_{k,m} = co - L_k \setminus \{a^m\}$. Hence, $co - L_k \cap co - L_{k,m} = co - L_{k,m}$.

We continue in showing $\mathcal{L} \notin MON - INF$. Suppose there is an IIM M which monotonically infers $\mathcal{L}$. Let i be any informant for $\{a\}^*$. Hence, there must be an x such that

$j_x = M(i_x)$ and $L(G_{j_x}) = L_1$. Let $k = max\{|s| \mid s \in i_x^+ \cup i_x^-\}$. Then, i_x will be extended to an informant $\tilde{i}$ for L_k. Since $L_k \in \mathcal{L}$, there has to exist a y such that $j_y = M(\tilde{i}_y)$ and $L(G_{j_y}) = L_k$. Let $m = max\{|s| \mid s \in \tilde{i}_y^+ \cup \tilde{i}_y^-\}$. Obviously, $\tilde{i}_y$ is an initial segment of an informant i_{fool} for $L_{k,m}$. It is easy to see that M either does not work monotonically or it fails to infer $L_{k,m}$ from i_{fool}. If M performs the described mind changes when inferring $L_{k,m}$ from i_{fool}, we have $a^m \in L_1 \cap L_{k,m}$, but $a^m \notin L_k \cap L_{k,m}$. This violates the monotonicity requirement.

Claim C. $SMON - INF \setminus SMON^d - INF \neq \emptyset$

By $\mathcal{L}_{fin}$ we denote the set of all finite languages over the alphabet $\Sigma = \{a\}$. Obviously, $\mathcal{L} \in SMON - INF$. It is easy to see that $\mathcal{L} \notin SMON^d - INF$.

Claim D. $SMON^d - INF \setminus SMON - INF \neq \emptyset$

We define $\mathcal{L} = L_1, L_2, ...$ as follows: $L_1 = \{a\}^*$ and $L_k = \{a^j \mid 1 \leq j \leq k\}$ for $k \geq 2$. It is easy to recognize that $\mathcal{L} \in SMON^d - INF$. On the other hand, $\mathcal{L} \notin LIM - TXT$, but $SMON - INF \subset LIM - TXT$ (cf. Lange and Zeugmann (1992A, 1993)). Hence $\mathcal{L} \notin SMON - INF$.

q.e.d.

Corollary 1.

(1) $FIN - INF \subset SMON^d - INF$

(2) $FIN - INF \subset SMON - INF$

Proof. By Proposition 1 we know that $FIN - INF \subseteq SMON - INF \cap SMON^d - INF$. Since $SMON - INF \# SMON^d - INF$, we immediately conclude $FIN - INF \neq SMON - INF$ as well as $FIN - INF \neq SMON^d - INF$.

q.e.d.

Next to, we combine the monotonicity constraints characterized in Definition 3 and Definition 5. This may help to obtain a better understanding of the relationship between monotonic language learning and other well–known types of language learning.

Definition 6. (Kapur, 1992) *Let $SMON^\& - INF$ denote the class of indexed families identifiable by an IIM which works strong–monotonically as well as dual strong–*

monotonically. The classes $MON^{\&}-INF$ and $WMON^{\&}-INF$ are analogously defined.

Theorem 2.

(1) $WMON^{\&} - INF = LIM - INF$

(2) $SMON - INF \subset MON^{\&} - INF$

(3) $SMON^{d} - INF \subset MON^{\&} - INF$

(4) $FIN - INF = SMON^{\&} - INF$

Proof. By applying the same arguments as in the proof of Theorem 1 one may show assertion (1). It is easy to see that $SMON - INF \subseteq MON^{\&} - INF$ as well as $SMON^{d} - INF \subseteq MON^{\&} - INF$. Together with assertion (3) of Theorem 1, we conclude (2) and (3). Finally, a closer look at Definition 3(A) and 5(A) directly yields assertion (4).

q.e.d.

The following picture summarizes the results presented above.

$$\begin{tikzcd}[column sep=small]
& WMON - INF^{\&} = WMON^{d} - INF = WMON - INF = LIM - INF \arrow[dl, dash] \arrow[dr, dash] & \\
MON^{d} - INF \arrow[dr, dash] & & MON - INF \arrow[dl, dash] \\
& MON^{\&} - INF \arrow[dl, dash] \arrow[dr, dash] & \\
SMON^{d} - INF \arrow[dr, dash] & & SMON - INF \arrow[dl, dash] \\
& SMON^{\&} - INF = FIN - INF &
\end{tikzcd}$$

FIGURE 1

All lines between identification types indicate inclusions in the sense that the upper type always properly contains the lower one. If two identification types are not connected by ascending lines then they are incomparable.

4. Characterization Theorems

In this section we give characterizations of all types of monotonic language learning from positive and negative data. Characterizations play an important role in that they lead to a deeper insight into the problem how algorithms performing the inference process may work (cf. e.g. Blum and Blum (1975), Wiehagen (1977, 1991), Angluin (1980), Zeugmann (1983), Jain and Sharma (1989)). Starting with the pioneering paper of Blum and Blum (1975), several theoretical frameworks have been used for characterizing identification 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 presented characterizations have been successfully applied for solving highly nontrivial problems in complexity theory. Moreover, up to now it remained open how to solve the same problems without using these characterizations. It seems that characterizations may help to get a deeper understanding of the theoretical framework where the concepts for characterizing identification types are borrowed from. The characterization for $SMON-TXT$ (cf. Lange and Zeugmann (1992B)) can be considered as further example along this line. This characterization has the following consequence. If $\mathcal{L} \in SMON-TXT$, then set inclusion in $\mathcal{L}$ is decidable (if one chooses an appropriate description of $\mathcal{L}$). On the other hand, Jantke (1991B) proved that, if set inclusion of pattern languages is decidable, then the family of all pattern languages may be inferred strong–monotonically from positive data. However, it remained open whether the converse is also true. Using our result, we see it is, i.e., if one can design an algorithm that learns the family of all pattern languages strong–monotonically from positive data, then set inclusion of pattern languages is decidable. This may show at least a promising way how to solve the open problem whether or not set inclusion of pattern languages is decidable.

Our first theorem characterizes $SMON-INF$ in terms of recursively generable finite positive and negative tell–tales. A family of finite sets $(P_j)_{j\in\mathbb{N}}$ is said to be recursively generable, iff there is a total effective procedure g which, on every input j, generates all elements of P_j and stops. If the computation of $g(j)$ stops and there is no output, then P_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 3. *Let $\mathcal{L}$ be an indexed family of recursive languages. Then: $\mathcal{L} \in SMON-INF$ if and only if there are a space of hypotheses $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ and recursively generable families $(\hat{P}_j)_{j\in\mathbb{N}}$ and $(\hat{N}_j)_{j\in\mathbb{N}}$ of finite sets such that*

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

(2) For all $j \in \mathbb{N}$, $\emptyset \neq \hat{P}_j \subseteq L(\hat{G}_j)$ and $\hat{N}_j \subseteq co-L(\hat{G}_j)$.

(3) For all $k,\ j \in \mathbb{N}$, if $\hat{P}_k \subseteq L(\hat{G}_j)$ as well as $\hat{N}_k \subseteq co-L(\hat{G}_j)$, then $L(\hat{G}_k) \subseteq L(\hat{G}_j)$.

Proof. Necessity: Let $\mathcal{L} \in SMON-INF$. Then there are an IIM M and a space of hypotheses $(G_j)_{j\in\mathbb{N}}$ such that M infers any $L \in \mathcal{L}$ strong–monotonically with respect to $(G_j)_{j\in\mathbb{N}}$. We proceed in showing how to construct $(\hat{G}_j)_{j\in\mathbb{N}}$. This will be done in two steps. In the first step, we define a space of hypotheses $(\tilde{G}_j)_{j\in\mathbb{N}}$ as well as corresponding recursively generable families $(\tilde{P}_j)_{j\in\mathbb{N}}$ and $(\tilde{N}_j)_{j\in\mathbb{N}}$ of finite sets where $\tilde{P}_j$ may be empty for some $j \in \mathbb{N}$. Afterwards, we define a procedure which enumerates a certain subset of $\tilde{\mathcal{G}}$.

First step: Let $c : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be Cantor's pairing function. For all $k,\ x \in \mathbb{N}$ we set $\tilde{G}_{c(k,x)} = G_k$. Obviously, it holds $range(\mathcal{L}) = L(\tilde{\mathcal{G}})$. Let i^k be the lexicographically ordered informant for $L(G_k)$, and let $x \in \mathbb{N}$.
We define:

$$\tilde{P}_{c(k,x)} = \begin{cases} range^+(i^k_y), & if \quad y = min\{z \mid z \leq x,\ M(i^k_z) = k,\ range^+(i^k_z) \neq \emptyset\} \\ \emptyset, & otherwise \end{cases}$$

If $\tilde{P}_{c(k,x)} = range^+(i^k_y) \neq \emptyset$, then we set $\tilde{N}_{c(k,x)} = range^-(i^k_y)$. Otherwise, we define $\tilde{N}_{c(k,x)} = \emptyset$.

Second step: The space of hypotheses $(\hat{G}_j)_{j\in\mathbb{N}}$ will be defined by simply striking off all grammars $\tilde{G}_{c(k,x)}$ with $\tilde{P}_{c(k,x)} = \emptyset$. In order to save readability, we omit the corresponding bijective mapping yielding the enumeration $(\hat{G}_j)_{j\in\mathbb{N}}$ from $(\tilde{G}_j)_{j\in\mathbb{N}}$. If $\hat{G}_j$ is referring to $\tilde{G}_{c(k,x)}$, we set $\hat{P}_j = \tilde{P}_{c(k,x)}$ as well as $\hat{N}_j = \tilde{N}_{c(k,x)}$.

We have to show that $(\hat{G}_j)_{j\in\mathbb{N}}$, $(\hat{N}_j)_{j\in\mathbb{N}}$, and $(\hat{P}_j)_{j\in\mathbb{N}}$ do fulfil the announced properties. Obviously, $(\hat{P}_j)_{j\in\mathbb{N}}$ and $(\hat{N}_j)_{j\in\mathbb{N}}$ are recursively generable families of finite sets. Furthermore, it is easy to see that $L(\hat{\mathcal{G}}) \subseteq range(\mathcal{L})$. In order to prove (1), it suffices to show that for every $L \in \mathcal{L}$ there is at least one $j \in \mathbb{N}$ with $L = L(\tilde{G}_j)$ and $\tilde{P}_j \neq \emptyset$. Let i^L be L's lexicographically ordered informant. Since M has to infer L from i^L, too, and $L \neq \emptyset$, there are $k,\ x \in \mathbb{N}$ such that $M(i^L_x) = k,\ L = L(G_k),\ range^+(i^L_x) \neq \emptyset$ as well as $M(i^L_y) \neq k$ for all $y < x$. From that we immediately conclude that $L = L(\tilde{G}_j)$ and that $\tilde{P}_j \neq \emptyset$ for $j = c(k,x)$. Due to our construction, property (2) is obviously fulfilled. It remains to show (3). Suppose $k,\ j \in \mathbb{N}$ such that $\hat{P}_k \subseteq L(\hat{G}_j)$ and $\hat{N}_k \subseteq co - L(\hat{G}_j)$. We have to show $L(\hat{G}_k) \subseteq L(\hat{G}_j)$. In accordance with our construction one can easily observe: There is a uniquely defined initial segment, say i^k_x, of the lexicographically ordered informant for $L(\hat{G}_k)$ such that $range(i^k_x) = \hat{P}_k \cup \hat{N}_k$. Furthermore, $M(i^k_x) = m$ with $L(\hat{G}_k) = L(G_m)$. Additionally, since $\hat{P}_j \subseteq L(\hat{G}_j)$ as well as $\hat{N}_k \subseteq co - L(\hat{G}_j)$, i^k_x is an initial segment of the lexicographically ordered informant i^j of $L(\hat{G}_j)$.

Since M infers $L(\hat{G}_j)$ from informant i^j, there exist $r,\ n \in \mathbb{N}$ such that $M(i^j_{x+r}) = n$ and $L(\hat{G}_j) = L(G_n)$. Moreover, M works strong–monotonically. Thus, by the transitivity of "$\subseteq$" we obtain $L(\hat{G}_k) \subseteq L(\hat{G}_j)$.

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

$M(i_x) =$ "Generate $\hat{P}_j$ and $\hat{N}_j$ for $j = 1, ..., x$ and test whether

(A) $\hat{P}_j \subseteq i^+_x \subseteq L(\hat{G}_j)$, and

(B) $\hat{N}_j \subseteq i^-_x \subseteq co - L(\hat{G}_j)$.

In case there is at least a j fulfilling the test, output the minimal one, and request the next input.

Otherwise output nothing and request the next input."

Since all of the $\hat{P}_k$ and $\hat{N}_k$ are uniformly recursively generable and finite, we see that M is an IIM. We have to show that it infers L. Let $z = \mu k[L = L(\hat{G}_k)]$. We claim that M converges to z. Consider $\hat{P}_1$, ..., $\hat{P}_z$ as well as $\hat{N}_1$, ..., $\hat{N}_z$. Then there must be an x such that $\hat{P}_z \subseteq i_x^+ \subseteq L(\hat{G}_z)$ and $\hat{N}_z \subseteq i_x^- \subseteq co - L(\hat{G}_z)$. That means, at least after having fed i_x to M, the machine M outputs a hypothesis. Moreover, since $\hat{P}_z \subseteq i_{x+r}^+ \subseteq L(\hat{G}_z)$ and $\hat{N}_z \subseteq i_{x+r}^- \subseteq co - L(\hat{G}_z)$ for all $r \in \mathbb{N}$, the IIM M never produces a guess $j > z$ on i_{x+r}. Suppose, M converges to $j < z$. Then we have $\hat{P}_j \subseteq i_{x+r}^+ \subseteq L(\hat{G}_j) \neq L(\hat{G}_z)$ and $\hat{N}_j \subseteq i_{x+r}^- \subseteq co - L(\hat{G}_j)$ for all $r \in \mathbb{N}$.

Case 1. $L(\hat{G}_z) \setminus L(\hat{G}_j) \neq \emptyset$

Consequently, there is at least one string $s \in L(\hat{G}_z) \setminus L(\hat{G}_j)$ such that $(s, +)$ has to appear sometimes in i, say in i_{x+r} for some r. Thus, $i_{x+r}^+ \not\subseteq L(\hat{G}_j)$, a contradiction.

Case 2. $L(\hat{G}_j) \setminus L(\hat{G}_z) \neq \emptyset$

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

Therefore, M converges to z from informant i. In order to complete the proof we show that M works strong–monotonically. Suppose that M sometimes outputs k and changes its mind to j in some subsequent step. Hence, $M(i_x) = k$ and $M(i_{x+r}) = j$, for some x, $r \in \mathbb{N}$. Due to the construction of M, we obtain $\hat{P}_k \subseteq i_x^+ \subseteq i_{x+r}^+ \subseteq L(\hat{G}_j)$ and $\hat{N}_k \subseteq i_x^- \subseteq i_{x+r}^- \subseteq co - L(\hat{G}_j)$. This yields $\hat{P}_k \subseteq L(\hat{G}_j)$ as well as $\hat{N}_k \subseteq co - L(\hat{G}_j)$. Finally, (3) implies $L(\hat{G}_k) \subseteq L(\hat{G}_j)$. Hence, M works indeed strong–monotonically.

q.e.d.

In turns out that we obtain a quite similar characterization for $SMON^d - INF$. The same proof technique presented above applies mutatis mutandis to prove Theorem 4.

Theorem 4. *Let $\mathcal{L}$ be an indexed family of recursive languages. Then: $\mathcal{L} \in SMON^d-INF$ if and only if there are a space of hypotheses $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ and recursively generable families $(\hat{P}_j)_{j\in\mathbb{N}}$ and $(\hat{N}_j)_{j\in\mathbb{N}}$ of finite sets such that*

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

(2) *For all* $j \in \mathbb{N}$, $\emptyset \neq \hat{P}_j \subseteq L(\hat{G}_j)$ *and* $\hat{N}_j \subseteq co - L(\hat{G}_j)$.

(3) *For all* $k,\ j \in \mathbb{N}$, *if* $\hat{P}_k \subseteq L(\hat{G}_j)$ *as well as* $\hat{N}_k \subseteq co - L(\hat{G}_j)$, *then* $co - L(\hat{G}_k) \subseteq co - L(\hat{G}_j)$.

Next to, we characterize $SMON^{\&} - INF$. Because $SMON^{\&} - INF = FIN - INF$, it suffices to present a characterization for $FIN - INF$. Note that a bit weaker theorem has been obtained independently by Mukouchi (1991). The difference is caused by Mukouchi's (1991) definition of finite identification from informant, since he demanded any indexed family $\mathcal{L}$ to be finitely learnt with respect to $\mathcal{L}$ itself. Therefore the problem arises whether or not this requirement might lead to a decrease in the inferring power. It does not, as we shall see.

However, even the next theorem has some special features distinguishing it from the characterizations already given. As pointed out above, dealing with characterizations has been motivated by the aim to elaborate a unifying approach to monotonic inference. Concerning $SMON - INF$ as well as $SMON^d - INF$ this goal has been completely met by showing that there is exactly one algorithm, i.e., that one described in Theorem 3 and Theorem 4, which can perform the desired inference task, if the space of hypotheses is appropriately chosen. The next theorem yields even a stronger implication. Namely, it shows, if there is a space of hypotheses at all such that $\mathcal{L} \in FIN - INF$ with respect to this space, then one can always use $\mathcal{L}$ itself as space of hypotheses, thereby again applying essentially one and the same inference procedure.

Theorem 5. *Let* $\mathcal{L}$ *be an indexed family of recursive languages. Then:* $\mathcal{L} \in FIN - INF$ *if and only if there are recursively generable families* $(P_j)_{j\in\mathbb{N}}$ *and* $(N_j)_{j\in\mathbb{N}}$ *of finite sets such that*

(1) *For all* $j \in \mathbb{N}$, $\emptyset \neq P_j \subseteq L_j$ *and* $N_j \subseteq co - L_j$.

(2) *For all* $k,\ j \in \mathbb{N}$, *if* $P_k \subseteq L_j$ *and* $N_k \subseteq co - L_j$, *then* $L_k = L_j$.

Proof. Necessity: Let $\mathcal{L} \in FIN - INF$. 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 proceed in showing how to construct $(P_j)_{j\in\mathbb{N}}$ and $(N_j)_{j\in\mathbb{N}}$. This is done in three steps. First, it does not seem likely, though conceivable that M produces its output before having received any pair $(s,+)$. Such a behavior might cause some technical trouble, since we aim to construct a family of non–empty tell–tales $(P_j)_{j\in\mathbb{N}}$. Therefore, we replace M by an IIM $\hat{M}$ as follows: On any input i_x, $\hat{M}$ simulates M on input i_x. If M produces a hypothesis on i_x, the IIM $\hat{M}$ additionally checks whether or not $i_x^+ \neq \emptyset$. In case it is, $\hat{M}$ outputs $M(i_x)$ and stops. Otherwise, $\hat{M}$ requests next input, until $i_{x+r}^+ \neq \emptyset$ for some $r \in \mathbb{N}$. Then it outputs $M(i_x)$ and stops. In particular, $\mathcal{L}$ is a family of non–empty languages. Thus, $\mathcal{L} \in FIN-INF(\hat{M})$, since $\mathcal{L} \in FIN-INF(M)$. Second, we construct $(\hat{P}_j)_{j\in\mathbb{N}}$ and $(\hat{N}_j)_{j\in\mathbb{N}}$ with respect to the space $\mathcal{G}$ of hypotheses. Third, we describe a procedure yielding the wanted families $(P_j)_{j\in\mathbb{N}}$ and $(N_j)_{j\in\mathbb{N}}$ with respect to $\mathcal{L}$.

Let $k \in \mathbb{N}$ be arbitrarily fixed. Furthermore, let i^k be the lexicographically ordered informant of $L(G_k)$. Since $\hat{M}$ finitely infers $L(G_k)$ from i^k, there exists an $x \in \mathbb{N}$ such that $\hat{M}(i_x^k) = m$ with $L(G_k) = L(G_m)$. We set $\hat{P}_k = range^+(i_x^k)$ and $\hat{N}_k = range^-(i_x^k)$. The desired families $(P_z)_{z\in\mathbb{N}}$ and $(N_z)_{z\in\mathbb{N}}$ are obtained as follows. Let $z \in \mathbb{N}$. In order to get P_z and N_z search for the least $j \in \mathbb{N}$ such that $\hat{P}_j \subseteq L_z$ and $\hat{N}_j \subseteq co-L_z$. Set $P_z = \hat{P}_j$ and $N_z = \hat{N}_j$. Note that, by construction, for every z at least one wanted j has to exist.

We have to show that $(P_j)_{j\in\mathbb{N}}$ and $(N_j)_{j\in\mathbb{N}}$ fulfil the announced properties. Due to our construction, property (1) holds obviously. It remains to show (2). Suppose $z,\ y \in \mathbb{N}$ such that $P_z \subseteq L_y$ and $N_z \subseteq co-L_y$. In accordance with our construction there is an index k such that $P_z = \hat{P}_k$ and $N_z = \hat{N}_k$. Moreover, due to construction there is an initial segment of the lexicographically ordered informant i^k of $L(G_k)$, say i_x^k, such that $range(i_x^k) = \hat{P}_k \cup \hat{N}_k$. Furthermore, $\hat{M}(i_x^k) = m$ with $L(G_k) = L(G_m)$. Since $\hat{P}_k \subseteq L_y$ and $\hat{N}_k \subseteq co-L_y$, i_x^k is an initial segment of some informant for L_y, too. Taking into account that $\hat{M}$ finitely infers L_y from any informant and that $\hat{M}(i_x^k) = m$, we immediately obtain $L_y = L(G_m)$. Finally, due to the definition of P_z and N_z we additionally know that $\hat{P}_k \subseteq L_z$ and $\hat{N}_k \subseteq co-L_z$, 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 that finitely infers any $L \in \mathcal{L}$

from any informant with respect to $\mathcal{L}$. So let $L \in \mathcal{L}$, let i be any informant for L, and $x \in \mathbb{N}$.

$M(i_x) =$ "Generate P_j and N_j for $j = 1, ..., x$ and test whether

(A) $P_j \subseteq i_x^+ \subseteq L_j$ and

(B) $N_j \subseteq i_x^- \subseteq co - L_j$.

In case there is at least a j fulfilling the test, output the minimal one and stop.

Otherwise, output nothing and request the next input."

Since all of the P_j and N_j are uniformly recursively generable and finite, we see that M is an IIM. We have to show that it finitely infers L. Let $j = \mu n[L = L_n]$. Then there must be an $x \in \mathbb{N}$ such that $P_j \subseteq i_x^+$ as well as $N_j \subseteq i_x^-$. That means, at least after having fed i_x to M, the machine M outputs a hypothesis and stops. Suppose M produces a hypotheses k with $k \neq j$ and stops. Hence, there has to be a z with $z < x$ such that $P_k \subseteq i_z^+$ and $N_k \subseteq i_z^-$. Since $z < x$, it follows $P_k \subseteq L_j$ and $N_k \subseteq co - L_j$. Hence, (2) implies $L_k = L_j$. Consequently, M outputs a correct hypothesis for L and stops afterwards.

q.e.d.

We continue in characterizing $MON - INF$ as well as $MON^d - INF$.

Theorem 6. *Let $\mathcal{L}$ be an indexed family of recursive languages. Then: $\mathcal{L} \in MON - INF$ if and only if there are a space of hypotheses $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ and recursively generable families $(\hat{P}_j)_{j\in\mathbb{N}}$ and $(\hat{N}_j)_{j\in\mathbb{N}}$ of finite sets such that*

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

(2) For all $j \in \mathbb{N}$, $\emptyset \neq \hat{P}_j \subseteq L(\hat{G}_j)$ and $\hat{N}_j \subseteq co - L(\hat{G}_j)$

(3) For all k, $j \in \mathbb{N}$, and for all $L \in \mathcal{L}$, if $\hat{P}_k \cup \hat{P}_j \subseteq L(\hat{G}_j) \cap L$ as well as $\hat{N}_k \cup \hat{N}_j \subseteq co - L(\hat{G}_j) \cap co - L$, then $L(\hat{G}_k) \cap L \subseteq L(\hat{G}_j) \cap L$.

Proof. Necessity: Let $\mathcal{L} \in MON-INF$. Then there are an IIM M and a space of hypotheses $(G_j)_{j\in\mathbb{N}}$ such that M infers any $L \in \mathcal{L}$ monotonically from any informant with respect to $(G_j)_{j\in\mathbb{N}}$. Without loss of generality, we can assume that M works conservatively, too, (cf. Lange and Zeugmann (1992A, 1993)). The space of hypotheses $(\hat{G}_j)_{j\in\mathbb{N}}$ as well as the corresponding recursively generable families $(\hat{P}_j)_{j\in\mathbb{N}}$ and $(\hat{N}_j)_{j\in\mathbb{N}}$ of finite sets are defined as in the proof of Theorem 1.

We proceed in showing that $(\hat{G}_j)_{j\in\mathbb{N}}$, $(\hat{N}_j)_{j\in\mathbb{N}}$, and $(\hat{P}_j)_{j\in\mathbb{N}}$ do fulfil the announced properties. By applying the same arguments as in the proof of Theorem 3 one obtains (1) and (2). It remains to show (3). Suppose $L \in \mathcal{L}$ and $k,\ j \in \mathbb{N}$ such that $\hat{P}_k \cup \hat{P}_j \subseteq L(\hat{G}_j) \cap L$ as well as $\hat{N}_k \cup \hat{N}_j \subseteq co-L(\hat{G}_j) \cap co-L$. We have to show $L(\hat{G}_k) \cap L \subseteq L(\hat{G}_j) \cap L$. Due to our construction, we can make the following observations. There is a uniquely defined initial segment of the lexicographically ordered informant i^k for $L(\hat{G}_k)$, say i^k_x, such that $range(i^k_x) = \hat{P}_k \cup \hat{N}_k$. Moreover, $M(i^k_x) = m$ with $L(\hat{G}_k) = L(G_m)$. By i^j_y we denote the uniquely defined initial segment of the lexicographically ordered informant i^j for $L(\hat{G}_j)$ with $range(i^j_y) = \hat{P}_j \cup \hat{N}_j$. Furthermore, $M(i^j_y) = n$ and $L(\hat{G}_j) = L(G_n)$. From $\hat{P}_k \subseteq L(\hat{G}_j)$ and $\hat{N}_k \subseteq co - L(\hat{G}_j)$, it follows $i^k_x \sqsubseteq i^j$. Since $\hat{P}_j \subseteq L$ and $\hat{N}_j \subseteq co - L$, we conclude that i^j_y is an initial segment of the lexicographically ordered informant i^L for L.

We have to distinguish the following three cases.

Case 1. $x = y$

Hence, $m = n$ and therefore $L(\hat{G}_k) = L(\hat{G}_j)$. This implies $L(\hat{G}_k) \cap L \subseteq L(\hat{G}_j) \cap L$.

Case 2. $x < y$

Now, we have $i^k_x \sqsubseteq i^j_y \sqsubseteq i^L$. Moreover, M monotonically infers L from informant i^L. By the transitivity of "$\subseteq$" we immediately obtain $L(\hat{G}_k) \cap L \subseteq L(\hat{G}_j) \cap L$.

Case 3. $y < x$

Hence, $i^j_y \sqsubseteq i^k_x \sqsubseteq i^j$. Since M works conservatively, too, it follows $m = n$. Therefore, $L(\hat{G}_k) = L(\hat{G}_j)$. This implies $L(\hat{G}_k) \cap L \subseteq L(\hat{G}_j) \cap L$.

Hence, $(\hat{G}_j)_{j\in\mathbb{N}}$, $(\hat{N}_j)_{j\in\mathbb{N}}$ as well as $(\hat{P}_j)_{j\in\mathbb{N}}$ have indeed the announced properties.

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

$M(i_x) =$ "Generate $\hat{P}_j$ and $\hat{N}_j$ for $j = 1, ..., x$ and test whether

(A) $\hat{P}_j \subseteq i_x^+ \subseteq L(\hat{G}_j)$ and

(B) $\hat{N}_j \subseteq i_x^- \subseteq co - L(\hat{G}_j)$.

In case there is at least a j fulfilling the test, output the minimal one and request the next input.

Otherwise, output nothing and request the next input."

Since all of the $\hat{P}_k$ and $\hat{N}_k$ are uniformly recursively generable and finite, we see that M is an IIM. We have to show that it infers L. Let $z = \mu k[L = L(\hat{G}_k)]$. We claim that M converges to z. Consider $\hat{P}_1$, ..., $\hat{P}_z$ as well as $\hat{N}_1$, ..., $\hat{N}_z$. Then there must be an x such that $\hat{P}_z \subseteq i_x^+ \subseteq L(\hat{G}_z)$ and $\hat{N}_z \subseteq i_x^- \subseteq co - L(\hat{G}_z)$. That means, at least after having fed i_x to M, the machine M outputs a hypothesis. Moreover, since $\hat{P}_z \subseteq i_{x+r}^+ \subseteq L(\hat{G}_z)$ as well as $\hat{N}_z \subseteq i_{x+r}^- \subseteq co - L(\hat{G}_z)$ for all $r \in \mathbb{N}$, the IIM M never produces a guess $j > z$ on i_{x+r}.

Suppose, M converges to $j < z$. Then we have: $\hat{P}_j \subseteq i_{x+r}^+ \subseteq L(\hat{G}_j) \neq L(\hat{G}_z)$ and $\hat{N}_j \subseteq i_{x+r}^- \subseteq co - L(\hat{G}_j)$ for all $r \in \mathbb{N}$.

Case 1. $L(\hat{G}_z) \setminus L(\hat{G}_j) \neq \emptyset$

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

Case 2. $L(\hat{G}_j) \setminus L(\hat{G}_z) \neq \emptyset$

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

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

Case 1. $L(\hat{G}_j) = L$
Hence, $L(\hat{G}_k) \cap L \subseteq L(\hat{G}_j) \cap L = L$ is obviously fulfilled.

Case 2. $L(\hat{G}_j) \neq L$
Due to the definition of M, it holds $\hat{P}_k \subseteq i_x^+ \subseteq i_{x+r}^+ \subseteq L(\hat{G}_j)$. Hence, $\hat{P}_k \subseteq L \cap L(\hat{G}_j)$. Furthermore, we have $\hat{N}_k \subseteq i_x^- \subseteq i_{x+r}^- \subseteq co-L(\hat{G}_j)$. This implies $\hat{N}_k \subseteq co-L(\hat{G}_j) \cap co-L$. Since $M(i_{x+r}) = j$, it holds that $\hat{P}_j \subseteq L$ and $\hat{N}_j \subseteq co-L$. This yields $\hat{P}_k \cup \hat{P}_j \subseteq L(\hat{G}_j) \cap L$ as well as $\hat{N}_k \cup \hat{N}_j \subseteq co-L(\hat{G}_j) \cap co-L$. From (3), we obtain $L(\hat{G}_k) \cap L \subseteq L(\hat{G}_j) \cap L$.

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

q.e.d.

Next we present the announced characterization of $MON^d - INF$.

Theorem 7. *Let $\mathcal{L}$ be an indexed family of recursive languages. Then: $\mathcal{L} \in MON^d - INF$ if and only if there are a space of hypotheses $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ and recursively generable families $(\hat{P}_j)_{j\in\mathbb{N}}$ and $(\hat{N}_j)_{j\in\mathbb{N}}$ of finite sets such that*

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

(2) For all $j \in \mathbb{N}$, $\emptyset \neq \hat{P}_j \subseteq L(\hat{G}_j)$ and $\hat{N}_j \subseteq co-L(\hat{G}_j)$

(3) For all $k,\ j \in \mathbb{N}$, and for all $L \in \mathcal{L}$, if $\hat{P}_k \cup \hat{P}_j \subseteq L(\hat{G}_j) \cap L$ as well as $\hat{N}_k \cup \hat{N}_j \subseteq co-L(\hat{G}_j) \cap co-L$, then $co-L(\hat{G}_k) \cap co-L \subseteq co-L(\hat{G}_j) \cap co-L$.

Proof. Necessity: Let $\mathcal{L} \in MON^d - INF$. Then there are an IIM M and a space of hypotheses $(G_j)_{j\in\mathbb{N}}$ such that M infers any $L \in \mathcal{L}$ dual monotonically from any informant with respect to $(G_j)_{j\in\mathbb{N}}$. First we claim that, without loss of generality, we can assume M working conservatively, too. This can be analogously seen as in Lange and Zeugmann (1992A, 1993). The space of hypotheses $(\hat{G}_j)_{j\in\mathbb{N}}$ as well as the corresponding recursively generable families $(\hat{P}_j)_{j\in\mathbb{N}}$ and $(\hat{N}_j)_{j\in\mathbb{N}}$ of finite sets are defined in the same way as in the proof of Theorem 3.

We proceed in showing that $(\hat{G}_j)_{j\in\mathbb{N}}$, $(\hat{N}_j)_{j\in\mathbb{N}}$, and $(\hat{P}_j)_{j\in\mathbb{N}}$ do fulfil the announced properties. By applying the same arguments as in the proof of Theorem 3 one obtains (1) and (2). It remains to show (3). Suppose $L \in \mathcal{L}$ and $k,\ j \in \mathbb{N}$ such that $\hat{P}_k \cup \hat{P}_j \subseteq$

$L(\hat{G}_j) \cap L$ as well as $\hat{N}_k \cup \hat{N}_j \subseteq co - L(\hat{G}_j) \cap co - L$. We have to show $co - L(\hat{G}_k) \cap co - L \subseteq co - L(\hat{G}_j) \cap co - L$. Due to our construction, we can make the following observations. There is a uniquely defined initial segment of the lexicographically ordered informant i^k for $L(\hat{G}_k)$, say i^k_x, such that $range(i^k_x) = \hat{P}_k \cup \hat{N}_k$. Moreover, $M(i^k_x) = m$ with $L(\hat{G}_k) = L(G_m)$. By i^j_y we denote the uniquely defined initial segment of the lexicographically ordered informant i^j for $L(\hat{G}_j)$ with $range(i^j_y) = \hat{P}_j \cup \hat{N}_j$. Furthermore, $M(i^j_y) = n$ and $L(\hat{G}_j) = L(G_n)$. From $\hat{P}_k \subseteq L(\hat{G}_j)$ and $\hat{N}_k \subseteq co - L(\hat{G}_j)$, it follows $i^k_x \sqsubseteq i^j$. Since $\hat{P}_j \subseteq L$ and $\hat{N}_j \subseteq co - L$, we conclude that i^j_y is an initial segment of the lexicographically ordered informant i^L for L.

We have to distinguish the following three cases.

Case 1. $x = y$

Hence, $m = n$ and therefore $L(\hat{G}_k) = L(\hat{G}_j)$. This implies $co - L(\hat{G}_k) \cap co - L \subseteq co - L(\hat{G}_j) \cap co - L$.

Case 2. $x < y$

Now, we have $i^k_x \sqsubseteq i^j_y \sqsubseteq i^L$. Moreover, M dual monotonically infers L from informant i^L. By the transitivity of "$\subseteq$" we immediately obtain that $co - L(\hat{G}_k) \cap co - L \subseteq co - L(\hat{G}_j) \cap co - L$.

Case 3. $y < x$

Hence, $i^j_y \sqsubseteq i^k_x \sqsubseteq i^j$. Since M works conservatively, too, it follows $m = n$. Therefore, $L(\hat{G}_k) = L(\hat{G}_j)$. This implies $co - L(\hat{G}_k) \cap co - L \subseteq co - L(\hat{G}_j) \cap co - L$.

Hence, $(\hat{G}_j)_{j\in\mathbb{N}}$, $(\hat{N}_j)_{j\in\mathbb{N}}$ as well as $(\hat{P}_j)_{j\in\mathbb{N}}$ have indeed the announced properties.

Sufficiency: It suffices to prove that there is an IIM M inferring any $L \in \mathcal{L}$ dual monotonically from any informant with respect to $\hat{\mathcal{G}}$. So let $L \in \mathcal{L}$, let i be any informant for L, and $x \in \mathbb{N}$.

$M(i_x) =$ "Generate $\hat{P}_j$ and $\hat{N}_j$ for $j = 1, ..., x$ and test whether

(A) $\hat{P}_j \subseteq i^+_x \subseteq L(\hat{G}_j)$ and

(B) $\hat{N}_j \subseteq i^-_x \subseteq co - L(\hat{G}_j)$.

In case there is at least a j fulfilling the test, output the minimal one and request the next input.

Otherwise, output nothing and request the next input."

By applying exactly the same arguments as in the proof of Theorem 6, we may conclude that M converges to $z = \mu k[L = L(\hat{G}_k)]$ from informant i. It remains to show that M works dual monotonically. Suppose M outputs k and changes its mind to j in some subsequent step. Consequently, $M(i_x) = k$ and $M(i_{x+r}) = j$, for some $x,\ r \in \mathbb{N}$.

Case 1. $L(\hat{G}_j) = L$

Hence, $co - L(\hat{G}_k) \cap co - L \subseteq co - L(\hat{G}_j) \cap co - L = co - L$ is obviously fulfilled.

Case 2. $L(\hat{G}_j) \neq L$

Due to the definition of M, it holds $\hat{P}_k \subseteq i_x^+ \subseteq i_{x+r}^+ \subseteq L(\hat{G}_j)$. Hence, $\hat{P}_k \subseteq L \cap L(\hat{G}_j)$. Furthermore, we have $\hat{N}_k \subseteq i_x^- \subseteq i_{x+r}^- \subseteq co{-}L(\hat{G}_j)$. This implies $\hat{N}_k \subseteq co{-}L(\hat{G}_j) \cap co{-}L$. Since $M(i_{x+r}) = j$, it holds that $\hat{P}_j \subseteq L$ and $\hat{N}_j \subseteq co{-}L$. This yields $\hat{P}_k \cup \hat{P}_j \subseteq L(\hat{G}_j) \cap L$ as well as $\hat{N}_k \cup \hat{N}_j \subseteq co - L(\hat{G}_j) \cap co - L$. From (3), we obtain $co - L(\hat{G}_k) \cap co - L \subseteq co - L(\hat{G}_j) \cap co - L$.

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

q.e.d.

Finally in this section, we characterize $MON^{\&} - INF$. Obviously, one may easily use property (3) of Theorem 6 and 7 to obtain a characterization of $MON^{\&} - INF$. However, such a characterization would neither be very useful in potential applications nor mathematically satisfactory. Instead, our new property (3) delivers easy to handle conditions that shed some additional light on the combination of monotonicity constraints. Note that a similar idea may be used to characterize the combination of monotonic and dual monotonic language learning from positive data (cf. Zeugmann, Lange and Kapur (1992)).

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

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

(2) *For all* $j \in \mathbb{N}$, $\emptyset \neq \hat{P}_j \subseteq L(\hat{G}_j)$ *and* $\hat{N}_j \subseteq co - L(\hat{G}_j)$

(3) *For all* $k,\ j \in \mathbb{N}$, *and for all* $L \in \mathcal{L}$, *if* $\hat{P}_k \cup \hat{P}_j \subseteq L(\hat{G}_j) \cap L$ *as well as* $\hat{N}_k \cup \hat{N}_j \subseteq co - L(\hat{G}_j) \cap co - L$, *then*

(i) $L(\hat{G}_j) \setminus L(\hat{G}_k) \subseteq L$

(ii) $(L(\hat{G}_k) \setminus L(\hat{G}_j)) \cap L = \emptyset$

Proof. Necessity: Let $\mathcal{L} \in MON^{\&} - INF$. Then there are an IIM M and a space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ of hypotheses such that $\mathcal{L} \subseteq MON^{\&} - INF(M)$ with respect to $\mathcal{G}$. Moreover, without loss of generality we may assume that M works conservatively and consistently (cf. Lange and Zeugmann (1992A)). The wanted space $(\hat{G}_j)_{j\in\mathbb{N}}$ as well as the corresponding recursively generable families $(\hat{P}_j)_{j\in\mathbb{N}}$ and $(\hat{N}_j)_{j\in\mathbb{N}}$ of finite sets are defined in the same way as in the proof of Theorem 3. Property (1) and (2) may be analogously proved as in the proof of Theorem 3. We omit the details. It remains to show that property (3) is fulfilled. Using the same arguments as in the proof of Theorem 6 and 7, one straightforwardly obtains:

(A) For all $k,\ j \in \mathbb{N}$, and for all $L \in \mathcal{L}$, if $\hat{P}_k \cup \hat{P}_j \subseteq L(\hat{G}_j) \cap L$ as well as $\hat{N}_k \cup \hat{N}_j \subseteq co - L(\hat{G}_j) \cap co - L$, then $L(\hat{G}_k) \cap L \subseteq L(\hat{G}_j) \cap L$.

(B) For all $k,\ j \in \mathbb{N}$, and for all $L \in \mathcal{L}$, if $\hat{P}_k \cup \hat{P}_j \subseteq L(\hat{G}_j) \cap L$ as well as $\hat{N}_k \cup \hat{N}_j \subseteq co - L(\hat{G}_j) \cap co - L$, then $co - L(\hat{G}_k) \cap co - L \subseteq co - L(\hat{G}_j) \cap co - L$.

Suppose, (i) is not fulfilled. Hence, there is a string $s \in L(\hat{G}_j) \setminus L(\hat{G}_k)$ and $s \notin L$. Thus, $s \in co-L(\hat{G}_k) \cap co-L$ but $s \notin co-L(\hat{G}_j) \cap co-L$. Therefore, $co-L(\hat{G}_k) \cap co-L \not\subseteq co - L(\hat{G}_j) \cap co - L$, a contradiction to (B). This proves (i) of property (3).

Next we show (ii). Suppose the converse, i.e., there is a string $s \in (L(\hat{G}_k) \setminus L(\hat{G}_j)) \cap L$. Then, $s \in L(\hat{G}_k) \cap L$ and $s \notin L(\hat{G}_j)$. Consequently, $s \notin L(\hat{G}_j) \cap L$. Summarizing, we obtain that $L(\hat{G}_k) \cap L \not\subseteq L(\hat{G}_j) \cap L$, a contradiction to (A). This proves (ii), and hence the necessity is shown.

Sufficiency: It suffices to prove that there is an IIM M simultaneously inferring any $L \in \mathcal{L}$ monotonically and dual monotonically on any informant with respect to $\hat{\mathcal{G}}$. So let i be any informant for L, and $x \in \mathbb{N}$.

$M(t_x) =$ "Generate $\hat{P}_j$ and $\hat{N}_j$ for $j = 1, ..., x$ and test whether

(A) $\hat{P}_j \subseteq i_x^+ \subseteq L(\hat{G}_j)$ and

(B) $\hat{N}_j \subseteq i_x^- \subseteq co - L(\hat{G}_j)$.

In case there is at least a j fulfilling the test, output the minimal one and request the next input.

Otherwise, output nothing and request the next input."

Using exactly the same arguments as in the proof of Theorem 6, one directly obtains that M converges on i to the least index z satisfying $L = L(\hat{G}_z)$. It remains to prove that M works monotonically as well as dual monotonically. Suppose M outputs k and changes its mind to j in some subsequent step.

Case 1. $L(\hat{G}_j) = L$

Hence, $L(\hat{G}_k) \cap L \subseteq L(\hat{G}_j) \cap L = L$ as well as $co - L(\hat{G}_k) \cap co - L \subseteq co - L(\hat{G}_j) \cap co - L = co - L$ are trivially satisfied.

Case 2. $L(\hat{G}_j) \neq L$

By definition of M we have: $\hat{P}_k \subseteq i_x^+ \subseteq i_{x+r}^+ \subseteq L(\hat{G}_j)$. Therefore, $\hat{P}_k \subseteq L(\hat{G}_j) \cap L$. Moreover, by construction we get $\hat{N}_k \subseteq i_x^- \subseteq i_{x+r}^- \subseteq co - L(\hat{G}_j)$, and hence, $\hat{N}_k \subseteq co - L(\hat{G}_j) \cap co - L$. Since $M(i_{x+r}) = j$, it holds that $\hat{P}_j \subseteq L$ and $\hat{N}_j \subseteq co - L$. Consequently, we obtain that $\hat{P}_k \cup \hat{P}_j \subseteq L(\hat{G}_j) \cap L$ as well as $\hat{N}_k \cup \hat{N}_j \subseteq co - L(\hat{G}_j) \cap co - L$. Applying property (3) we conclude that

(a) $L(\hat{G}_j) \setminus L(\hat{G}_k) \subseteq L$

(b) $(L(\hat{G}_k) \setminus L(\hat{G}_j)) \cap L = \emptyset$

Suppose, M does not work monotonically. Hence, $L(\hat{G}_k) \cap L \not\subseteq L(\hat{G}_j) \cap L$. Consequently, there is a string $s \in L(\hat{G}_k) \cap L$ satisfying $s \notin L(\hat{G}_j) \cap L$. Since $s \in L$, we immediately get $s \notin L(\hat{G}_j)$. Thus, there is a string $s \in (L(\hat{G}_k) \setminus L(\hat{G}_j)) \cap L$, and hence, (b) is contradicted. Therefore, M works indeed monotonically.

Suppose, M does not work dual monotonically. Consequently, $co - L(\hat{G}_k) \cap co - L \not\subseteq co - L(\hat{G}_j) \cap co - L$. Hence, there is a string $s \in co - L(\hat{G}_k) \cap co - L$ fulfilling $s \notin co - L(\hat{G}_j) \cap co - L$. Thus, $s \notin L(\hat{G}_k)$. Moreover, since $s \in co - L$ and $s \notin co - L(\hat{G}_j) \cap co - L$, we get that $s \notin L$ as well as $s \in L(\hat{G}_j)$. Hence, there is a string $s \in L(\hat{G}_j) \setminus L(\hat{G}_k)$ satisfying $s \notin L$, a contradiction to (a). Therefore, M also works dual monotonically.

q.e.d.

Since $WMON - INF = WMON^d - INF = WMON^\& - INF = LIM - INF$ and because of the following trivial proposition, there is no need at all for characterizing any type of weak–monotonic language learning from informant. It can be easily shown that any appropriate IIM working in accordance with the *identification by enumeration principle* is able to infer every indexed family of recursive languages from informant.

Proposition 2. *For any indexed family $\mathcal{L}$ of recursive languages we have $\mathcal{L} \in LIM - INF$.*

5. Conclusions

We have characterized strong–monotonic, monotonic, and weak–monotonic language learning as well as the corresponding types of dual monotonic language learning from positive and negative data. All these characterization theorems lead to a deeper insight into the problem what actually may be inferred monotonically. It turns out that each of these inference tasks can be performed by applying exactly the same learning algorithm.

Next we point out another interesting aspect of Angluin's (1980) as well as of our characterizations. Freivalds, Kinber and Wiehagen (1989) 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 argument/value-pairs containing at least the 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 *exactly* as powerful as behaviorally correct identification. The same approach may be undertaken in language learning (cf.

Lange and Wiehagen (1991)). Now it is not hard to prove that any indexed family $\mathcal{L}$ can be finitely inferred from good examples, where for each $L \in \mathcal{L}$ any superset of any of L's tell–tales may serve as good example.

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 of recursive languages.

Acknowledgement

The authors gratefully acknowledge many enlightening discussions with Rolf Wiehagen concerning the characterization of learning algorithms. The first author has been partially supported by the German Ministry for Research and Technology (BMFT) within the joint project GOSLER under grant no. 01 IW 101.

6. References

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

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

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

[4] Blum, L. and Blum, M. (1975) Toward a mathematical theory of inductive inference, *Information and Control*, 28:122 - 155.

[5] Bucher, W. and Maurer, H. (1984) *Theoretische Grundlagen der Programmiersprachen, Automaten und Sprachen* (Zürich: Bibliographisches Institut AG, Wissenschaftsverlag).

[6] Case, J. (1988) The Power of Vacillation, In D. Haussler and L. Pitt (eds.) *Proc. 1st Workshop on Computational Learning Theory* (Los Alto, CA: Morgan Kaufmann Publishers Inc.), 196 -205.

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

[8] Freivalds, R., Kinber, E. B. and Wiehagen, R. (1989) Inductive inference from good examples, In K.P. Jantke (ed.) *Proc. 2nd International Workshop on Analogical and Inductive Inference*, Lecture Notes in Artificial Intelligence 397 (Berlin: Springer-Verlag), 1 - 17.

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

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

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

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

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

[14] Jantke, K.P. (1991B) Monotonic and non–monotonic inductive inference of functions and patterns, In J.Dix, K.P. Jantke and P.H. Schmitt (eds.) *Proc. 1st International Workshop on Nonmonotonic and Inductive Logic*, Lecture Notes in Artificial Intelligence 543 (Berlin: Springer-Verlag), 161 - 177.

[15] Kapur, S. (1992) Monotonic language learning, In *Proc. 3rd Workshop on Algorithmic Learning Theory* (Tokyo: Ohmsha Ltd), to appear.

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

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

[18] Lange, S. and Zeugmann, T. (1992A) On the power of monotonic language learning, GOSLER–Report 05/92, Fachbereich Mathematik und Informatik, TH Leipzig.

[19] Lange, S. and Zeugmann, T. (1992B) Types of monotonic language learning and their characterizations, In *Proc. 5th Annual ACM Workshop on Computational Learning Theory* (New York: ACM Press), 377 - 390.

[20] Lange, S. and Zeugmann, T. (1992C) A unifying approach to monotonic language learning, In K.P. Jantke (ed.) *Proc. 3rd International Workshop on Analogical and Inductive Inference*, Lecture Notes in Artificial Intelligence 642 (Berlin: Springer-Verlag), 244 - 259.

[20] Lange, S., Zeugmann, T. and Kapur, S. (1992) Class preserving monotonic and dual monotonic language learning, submitted to Theoretical Computer Science.

[21] Mukouchi, Y. (1991) Definite inductive inference as a successful identification criterion, Research Institute of Fundamental Information Science, Kyushu University 33, Fukuoka, RIFIS-TR-CS-52.

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

[23] Solomonoff, R. (1964) A formal theory of inductive inference, *Information and Control*, 7:1 - 22, 234 - 254.

[24] Trakhtenbrot, B.A. and Barzdin, Ya.M. (1970) *Konetschnyje Awtomaty (Powedenie i Sintez)* (Moskwa: Nauka), (in Russian)
english translation: *Finite Automata–Behavior and Synthesis*, Fundamental Studies in Computer Science Vol.1 (Amsterdam: North–Holland), 1973.

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

[26] Wiehagen, R. (1977) Identification of formal languages, In J. Gruska (ed.) *Proc. Mathematical Foundations of Computer Science*, Lecture Notes in Computer Science 53 (Berlin: Springer-Verlag), 571 - 579.

[27] Wiehagen, R. (1991) A thesis in inductive inference, In J.Dix, K.P. Jantke and P.H. Schmitt (eds.), *Proc. 1st International Workshop on Nonmonotonic and Inductive Logic*, Lecture Notes in Artificial Intelligence 543 (Berlin: Springer-Verlag), 184 - 207.

[28] Wiehagen, R. and Liepe, W. (1976) Charakteristische Eigenschaften von erkennbaren Klassen rekursiver Funktionen, *Journal of Information Processing and Cybernetics (EIK)*, 12: 421 - 438.

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

[30] Zeugmann, T., Lange, S. and Kapur, S. (1992) Characterizations of class preserving monotonic and dual monotonic language learning, IRCS Report 92–24, The Institute for Research in Cognitive Science, University of Pennsylvania.

figure 1 on 12 page

$$\begin{tikzcd}[column sep=small]
& WMON-INF^{\&} = WMON^{d}-INF = WMON-INF = LIM-INF \arrow[dl, dash] \arrow[dr, dash] & \\
MON^{d}-INF \arrow[dr, dash] & & MON-INF \arrow[dl, dash] \\
& MON^{\&}-INF \arrow[dl, dash] \arrow[dr, dash] & \\
SMON^{d}-INF \arrow[dr, dash] & & SMON-INF \arrow[dl, dash] \\
& SMON^{\&}-INF = FIN-INF &
\end{tikzcd}$$