A Unifying Approach to Monotonic Language Learning on Informant

Steffen Lange*
TH Leipzig
FB Mathematik und Informatik
PF 66
O–7030 Leipzig
steffen@informatik.th-leipzig.de

Thomas Zeugmann
TH Darmstadt
Institut für Theoretische Informatik
Alexanderstr. 10
W–6100 Darmstadt
zeugmann@iti.informatik.th-darmstadt.de

Abstract

The present paper deals with strong-monotonic, monotonic and weak-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.

We characterize strong-monotonic, monotonic, weak-monotonic and 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 of 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 today 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

*This research has been supported by the German Ministry for Research and Technology (BMFT) under grant no. 01 IW 101.

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., the learner is fed with 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. Subsequently we have adopted their definitions to the inference of formal languages (cf. Lange and Zeugmann (1991, 1992A, 1992B)). The main underlying question can be posed as follows: Would it be possible to infer the unknown language in such a way that *only* better and better hypotheses are inferred?

The strongest interpretation of this requirement means that we are forced to produce an augmenting chain of languages, i.e., $L_i \subseteq L_j$ iff L_j is guessed later than L_i (cf. Definition 3 (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* reflects.

The third version of monotonicity, which we call weak–monotonicity, is derived from non–monotonic logics and adopts the concept of cumulativity. Hence, we only require $L_i \subseteq L_j$ as long as there are no data fed to the inference device after having produced L_i that contradict L_i (cf. Definition 3 (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 language learning one to the other as well as to previously studied modes of inference. This question has been completely answered in Lange and Zeugmann (1991, 1992A). In particular, weak–monotonically working learning devices are exactly as powerful as *conservatively* working ones. 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 on informant, one may conjecture that the characterizations for montonic 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 on 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 monotonic language learning, we present characterizations of monotonic language learning on informant in terms of recursively generable finite sets. In doing so, we will show that there is exactly one learning algorithm performing each of the desired inference tasks on 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 $N = \{1, 2, 3, ...\}$ we denote the set of all natural numbers. In the sequel we assume familarity 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 $w \in \Sigma^*$ is denoted by $|w|$. 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 $t = s_1, s_2, s_3, ...$ a sequence of strings from Σ^* such that $range(t) = \{s_k \mid k \in N\} = L$. Then t is said to be a *text* for L or, synonymously, a *positive presentation.* Furthermore, let $i = (s_1, b_1), (s_2, b_2), ...$ be a sequence of elements of $\Sigma^* \times \{+, -\}$ such that $range(i) = \{s_k \mid k \in N\} = \Sigma^*$, $i^+ = \{s_k \mid (s_k, b_k) = (s_k, +), k \in N\} = L$ and $i^- = \{s_k \mid (s_k, b_k) = (s_k, -), k \in N\} = co - L$. 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, respectively, e.g., $i_3 = (s_1, b_1), (s_2, b_2), (s_3, b_3)$. Let t be a text and let $x \in N$. We write t_x^+ as an abbreviation for $range^+(t_x) := \{s_k \mid k \leq x\}$. Furthermore, 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 $L_1, L_2, L_3, \ldots$ 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 $w \in \Sigma^*$ we have

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

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 a text t (an informant i) and it either requires the next input string, or it first outputs a hypothesis, i.e., a number encoding a certain computer program, and then it requires the next input string (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 with respect to the language to be learnt, i.e., to require containement of languages as described in the introduction. As it turned out, the latter approach increases considerably the power of monotonic language learning (cf. e.g. the example presented in the introduction). Furthermore, since we exclusively deal with indexed families $\mathcal{L} = (L_j)_{j \in N}$ of recursive languages we almost always take as space of hypotheses an enumerable family of grammars $G_1, G_2, G_3, \ldots$ over the terminal alphabet Σ satisfying $\mathcal{L} = \{L(G_j) \mid j \in N\}$. Moreover, we require that membership in $L(G_j)$ is uniformly decidable for all $j \in N$ and all strings $w \in \Sigma^*$. 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 $(j_x)_{x\in N}$ 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 $(G_j)_{j\in N}$ be a space of hypotheses. An IIM M $LIM-TXT$ $(LIM-INF)$–identifies L on a text t (an informant i) iff it almost always outputs a hypothesis and the sequence $(M(t_x))_{x\in N}$ $((M(i_x))_{x\in N})$ converges in the limit to a number j such that $L = L(G_j)$.*
Moreover, M $LIM-TXT$ $(LIM-INF)$–identifies L, iff M $LIM-TXT$ $(LIM-INF)$–identifies L on every text (informant) for L. We set:
$LIM-TXT(M) = \{L \in \mathcal{L} \mid M\ LIM-TXT-identifies\ L\}$ and define $LIM-INF(M)$ analogously.
Finally, let $LIM-TXT$ $(LIM-INF)$ denote the collection of all families $\mathcal{L}$ of indexed families of recursive languages for which there is an IIM M such that $\mathcal{L} \subseteq LIM-TXT(M)$ $(\mathcal{L} \subseteq LIM-INF(M))$.

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 wether or not an IIM has succesfully 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 $(G_j)_{j\in N}$ be a space of hypotheses. An IIM M $FIN-TXT$ $(FIN-INF)$–identifies L on a text t (an informant i) iff it outputs only a single and correct hypothesis j, i.e., $L = L(G_j)$, and stops.*
Moreover, M $FIN-TXT$ $(FIN-INF)$–identifies L, iff M $FIN-TXT$ $(FIN-INF)$–identifies L on every text (informant) for L. We set:
$FIN-TXT(M) = \{L \in \mathcal{L} \mid M\ FIN-TXT-identifies\ L\}$ and define $FIN-INF(M)$ analogously.

The resulting identification type is denoted by $FIN-TXT$ $(FIN-INF)$. 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 text (informant)*

(A) strong–monotonically

(B) monotonically

(C) weak–monotonically

iff

M $LIM-TXT$ $(LIM-INF)$–identifies L 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 $k \geq 1, k \in N$, 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* $L(G_{j_x}) \subseteq L(G_{j_{x+k}})$*).*

We denote by $SMON - TXT$, $SMON - INF$, $MON - TXT$, $MON - INF$, $WMON - TXT$, $WMON - INF$ the family 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 text t or informant i, respectively.

Note that even $SMON - TXT$ contains interesting "natural" families of formal languagues (cf. Lange and Zeugmann (1991, 1992A)). Finally in this section we define *conservatively* working IIMs.

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

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

(2) *If M on input* t_x *makes the guess* j_x *and then makes the guess* $j_{x+k} \neq j_x$ *at some subsequent step, then* $L(G_{j_x})$ *must fail to contain some string from* t_{x+k} *(*$L(G_{j_x})$ *must fail either to contain some string* $w \in i^+_{x+k}$ *or it generates some string* $w \in i^-_{x+k}$*).*

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

Intuitively speaking, a conservatively working IIM performs *exclusively* justified mind changes. Note that $WMON - TXT =$ *CONSERVATIVE–TXT* as well as $WMON - INF =$ *CONSERVATIVE–INF*. Finally, the figure below summarizes the known results concerning monotonic inference (cf. Lange and Zeugmann (1991, 1992A)).

$$\begin{array}{ccccccccc}
FIN - TXT & \subset & SMON - TXT & \subset & MON - TXT & \subset & WMON - TXT & \subset & LIM - TXT \\
\cap & \# & \cap & \# & \cap & \# & \cap & & \cap \\
FIN - INF & \subset & SMON - INF & \subset & MON - INF & \subset & WMON - INF & = & LIM - INF
\end{array}$$

(* # denotes incomparability of sets. *)

3. Characterization Theorems

In this section we give characterizations of strong–monotonic, monotonic and weak–monotonic inference from positive and negative data as well as for $FIN - INF$. 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 framworks 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 succesfully applied for solving highly nontrivial problems in complexity theory. Moreover, up to now it remains 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 wether 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 $(P_j)_{j\in N}$ 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 $\{L(G_j) \mid j \in N\}$ for any space $\mathcal{G} = (G_j)_{j\in N}$ of hypotheses.

Theorem 1. *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 $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in N}$ and recursively generable families $(\hat{P}_j)_{j\in N}$ and $(\hat{N}_j)_{j\in N}$ of finite sets such that*

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

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

(3) For all k, $j \in N$, if $\hat{P}_k \subset \hat{P}_j$ 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 $(G_j)_{j\in N}$ such that M infers any $L \in \mathcal{L}$ strong-monotonically with respect to $(G_j)_{j\in N}$. We proceed in showing how to construct $(\hat{G}_j)_{j\in N}$. This will be done in two steps. In the first step, we define a space of hypotheses $(\tilde{G}_j)_{j\in N}$ as well as corresponding recursively generable families $(\tilde{P}_j)_{j\in N}$ and $(\tilde{N}_j)_{j\in N}$ of finite sets where $\tilde{P}_j$ may be empty for some $j \in N$. Afterwards, we define a procedure which enumerates a certain subset of $\tilde{\mathcal{G}}$.

First step: Let $c : N \times N \to N$ be Cantor's pairing function. For all $k,\ x \in N$ 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 $x \in N$.
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 $(\hat{G}_j)_{j \in N}$ 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 enumaration $(\hat{G}_j)_{j \in N}$ from $(\tilde{G}_j)_{j \in N}$. 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 $(\hat{G}_j)_{j \in N}$, $(\hat{N}_j)_{j \in N}$, and $(\hat{P}_j)_{j \in N}$ do fulfil the announced properties. Obviously, $(\hat{P}_j)_{j \in N}$ and $(\hat{N}_j)_{j \in N}$ 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 $j \in N$ with $L = L(\tilde{G}_j)$ and $P_j \neq \emptyset$. Let i^L be L's lexicographically ordered informant. Since M has to infer L on i^L, too, and $L \neq \emptyset$, there are $k,\ x \in N$ 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 $k,\ j \in N$ 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 $L(\hat{G}_j)$ on informant i^j, there exist $r,\ n \in N$ 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 $L \in \mathcal{L}$ on any informant with respect to $\hat{\mathcal{G}}$. So let $L \in \mathcal{L}$, let i be any informant for L, and let $x \in N$.

$M(i_x) =$ "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) $\hat{N}_j \subseteq i^-_x \subset co - L(\hat{G}_j)$. 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 $k = \mu z[L = L(\hat{G}_z)]$. We claim that M converges to k. Consider $\hat{P}_1,\ ...,\ \hat{P}_k$ as well as $\hat{N}_1,\ ...,\ \hat{N}_k$. Then there must be an x

such that $\hat{P}_k \subseteq i_x^+ \subseteq L(\hat{G}_k)$ and $\hat{N}_k \subseteq i_x^- \subseteq co - L(\hat{G}_k)$. That means, at least after having fed i_x to M, the machine M outputs an hypothesis. Moreover, since $\hat{P}_k \subseteq i_{x+r}^+ \subseteq L(\hat{G}_k)$ and $\hat{N}_k \subseteq i_{x+r}^- \subseteq co - L(\hat{G}_k)$ for all $r \in N$, the IIM M never produces a guess $j > k$ on i_{x+r}. Suppose, M converges to $j < k$. Then we have $\hat{P}_j \subseteq i_{x+r}^+ \subseteq L(\hat{G}_j) \neq L(\hat{G}_k)$ and $\hat{N}_j \subseteq i_{x+r}^- \subseteq co - L(\hat{G}_j)$ for all $r \in N$.

Case 1: $L(\hat{G}_k) \setminus L(\hat{G}_j) \neq \emptyset$
Consequently, there is at least one string $s \in L(\hat{G}_k) \setminus L(\hat{G}_j)$ 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: $L(\hat{G}_j) \setminus L(\hat{G}_k) \neq \emptyset$
Then we may restrict ourselves to the case $L(\hat{G}_k) \subset L(\hat{G}_j)$, since otherwise we are again in case 1. Consequently, there is at least one string $s \in L(\hat{G}_j) \setminus L(\hat{G}_k)$ 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 k on informant i. In order to complete the proof we show that M works stong-monotonically. Suppose that M outputs k sometime and changes its mind to j in some subsequent step. Hence, $M(i_x) = k$ and $M(i_{x+r}) = j$, for some $x,\ r \in N$. 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.

Although, there are remarkable differences between formal language learning ¿from positive data, on the one hand, and language learning from positive and negative data, on the other hand, the characterizations of $SMON - INF$ is formally quite similar to that one of $SMON - TXT$, (cf. Lange and Zeugmann (1992B)).

Theorem 2. *Let $\mathcal{L}$ be an indexed family of recursive languages. Then: $\mathcal{L} \in SMON - TXT$ if and only if there are a space of hypotheses $\hat{\mathcal{G}} = (\hat{G}_j)_{j \in N}$ and a recursively generable family $(\hat{T}_j)_{j \in N}$ of finite and non–empty sets such that*

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

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

(3) For all $j,\ z \in N$, if $\hat{T}_j \subseteq L(\hat{G}_z)$, then $L(\hat{G}_j) \subseteq L(\hat{G}_z)$.

Surprisingly enough, the characterizations of $MON - INF$ and $MON - TXT$ are quite different. This is caused by the following facts. For characterizing $MON - TXT$, one has to construct a recursively generable family of finite tell-tales that should contain both, information concerning the corresponding language as well as concerning possible intersections of this language L with 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 *length* of the textsegment fed to M 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. In Lange and Zeugmann (1992B), a method is pointed out for overcoming these difficulties.

The problem explained above does not appear when characterizing $MON - INF$. For language learning on informant, one can make the following observation. Any IIM M performing a desired inference task can be always simulated by an *order-independent* IIM M' of the same inference power as M (cf. Blum and Blum (1975)). M' has simply to rearrange the incoming data in lexicographical order. Then, M' takes the longest initial segment of the rearranged informant which forms an initial segment of the lexicographically ordered informant of the language to be learnt as input and outputs the same hypothesis as M will do when proccesing this amount of data. This is quite different from what one can expect in language learning from positive data, since an IIM can never be sure to have already seen a complete initial segment of the lexicographically ordered text of a language to be learnt.

Next to, we present a characterization of $MON - INF$.

Theorem 3. *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 $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in N}$ and recursively generable families $(\hat{P}_j)_{j\in N}$ and $(\hat{N}_j)_{j\in N}$ of finite sets such that*

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

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

(3) For all $k,\ j \in N$, 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 $(G_j)_{j\in N}$ such that M infers any $L \in \mathcal{L}$ monotonically on any informant with respect to $(G_j)_{j\in N}$. Without loss of generality, we can assume that M works conservatively, too, (cf. Lange and Zeugmann (1991, 1992)). The space of hypotheses $(\hat{G}_j)_{j\in N}$ as well as the corresponding recursively generable families $(\hat{P}_j)_{j\in N}$ and $(\hat{N}_j)_{j\in N}$ of finite sets are defined as in the proof of Theorem 1.

We proceed in showing that $(\hat{G}_j)_{j\in N}$, $(\hat{N}_j)_{j\in N}$, and $(\hat{P}_j)_{j\in N}$ do fulfil the announced properties. By applying the same arguments as in the proof of Theorem 1 one obtains (1) and (2). It remains to show (3). Suppose $L \in \mathcal{L}$ and $k,\ j \in N$ 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 $i_x^k \sqsubseteq i_y^j \sqsubseteq i^L$. Moreover, M monotonically infers L on 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, $i_y^j \sqsubseteq i_x^k \sqsubseteq i^j$. 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, $(\hat{G}_j)_{j \in N}$, $(\hat{N}_j)_{j \in N}$ as well as $(\hat{P}_j)_{j \in N}$ have indeed the announced properties.

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

$M(t_x) =$ "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) $\hat{N}_j \subseteq i_x^- \subset co - L(\hat{G}_j)$.

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

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

Case 1: $L(\hat{G}_k) \setminus L(\hat{G}_j) \neq \emptyset$
Consequently, there is at least one string $s \in L(\hat{G}_k) \setminus L(\hat{G}_j)$ 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: $L(\hat{G}_j) \setminus L(\hat{G}_k) \neq \emptyset$
Then we may restrict ourselves to the case $L(\hat{G}_k) \subset L(\hat{G}_j)$, since otherwise we are again in case 1. Consequently, there is at least one string $s \in L(\hat{G}_j) \setminus L(\hat{G}_k)$ 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 k on 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, $r \in N$.

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.

¿From our characterization for $MON - INF$, it immediately follows that any family $\mathcal{L}$ which is inferrable on informant by a monotonically working IIM M can be learnt by a *rearrangment-independently* as well as monotonically working IIM M', too. Osherson, Stob and Weinstein (1986) defined *rearrangement-independent* IIMs as follows: An IIM M is rearrangement-independent iff its output depends only on the range and the length of its input. If we are dealt exclusively with monotonic inference from positive data by rearrangment independent IMMs, denoted by $MONR - TXT$, we obtain a quite similar characterization (cf. Lange and Zeugmann (1992B)).

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

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

(2) For all $j \in N$, $\hat{T}_j \subseteq L(\hat{G}_j)$.

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

(4) For all k, $j \in N$, and for all $L \in \mathcal{L}$, if $L(\hat{G}_j) \neq L \neq L(\hat{G}_k)$ and $\hat{T}_k \subseteq L(\hat{G}_j) \cap L$ as well as $\hat{T}_j \subseteq L$, then $L(\hat{G}_k) \cap L \subseteq L(\hat{G}_j) \cap L$.

Because of $WMON - INF = LIM - INF$ as well as the following trivial proposition, there is no need at all for characterizing $WMON - INF$. It can be easily shown that an appropriate *identification by enumeration strategy* is able to infer every indexed family of recursive languages on informant.

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

Finally, we present a characterization of $FIN - INF$. Note that an analogous theorem has been obtained independently by Mukouchi (1991).

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 $MON - INF$ as well as $SMON - INF$ this goal has been completely met by showing that there is exactly one algorithm, i.e. that one described in Theorem 1 and Theorem 3, which can perform the desired inference task, if the space of hypotheses is appropriately chosen. Obviously, the same algorithms can be applied for weak-monotonic inference, if the corresponding recursively generable families of finite sets will be appropriately choosen. 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 $(P_j)_{j\in N}$ and $(N_j)_{j\in N}$ of finite sets such that*

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

(2) For all k, $j \in N$, 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 $\mathcal{G} = (G_j)_{j\in N}$ 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 $(P_j)_{j\in N}$ and $(N_j)_{j\in N}$. This is done in two steps. First we construct $(\hat{P}_j)j \in N$ and $(\hat{N}_j)_{j\in N}$ with respect to the space $\mathcal{G}$ of hypotheses. Then we describe a procedure yielding the wanted families $(P_j)_{j\in N}$ and $(N_j)_{j\in N}$ with respect to $\mathcal{L}$.

Let $k \in N$ be arbitrarily fixed. Furthermore, let i^k be the lexicographically ordered informant of $L(G_k)$. Since M infers $L(G_k)$ finitely on i^k, there exists a $x \in N$ such that $M(i^k_x) = m$ with $L(G_k) = L(G_m)$. We set $\hat{P}_k = range^+(i^k_x)$ and $\hat{N}_k = range^-(i^k_x)$. The desired families $(P_j)_{j\in N}$ and $(N_j)_{j\in N}$ are obtained as follows. Let $z \in N$. In order to get P_z and N_z search for the least $j \in N$ 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 at least one wanted j has to exist, since for any pair $(\hat{P}_k, \hat{N}_k)$ of sets there is an informant i of some language $L \in \mathcal{L}$ such that $\hat{P}_k \subseteq i^+$ and $\hat{N}_k \subseteq i^-$.

We have to show that $(P_j)_{j\in N}$ and $(N_j)_{j\in N}$ fulfil the announced properties. Due to our construction, property (1) holds obviously. It remains to show (2). Suppose z, $y \in N$ 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^k_x, such that $range(i^k_x) = \hat{P}_k \cup \hat{N}_k$. Furthermore, $M(i^k_x) = m$ with $L(G_k) = L(G_m)$. Since $\hat{P}_k \subseteq L_y$ and $\hat{N}_k \subseteq co - L_y$, i^k_x is an initial segment of some informant for L_y, too. Taking into account that M finitely infers L_y on any informant and that $M(i^k_x) = m$, 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 inferring any $L \in \mathcal{L}$ finitely on any informant with respect to $\mathcal{L}$. So let $L \in \mathcal{L}$, let i be any informant for L, and $x \in N$.

$M(t_x) =$ "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 request the next input.

Otherwise, output nothing and request the next input."

Since all of the P_j and N_j are uniformly recursively generable and finite, we see that M is an IIM. We have to show that it infers L. Let $j = \mu n[L = L_n]$. Then there must be an $x \in N$ 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 an 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 $L_k = L_j$. Concequently, M outputs a correct hypotheses for L and stops afterwards.

q.e.d.

4. Conclusions

We have characterized strong-monotonic, monotonic, and weak-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 identification in the limit. 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 languague learning have special features distinguishing them from monotonic inference of recusive 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.

5. References

[1] Angluin, D., (1980), Inductive Inference of Formal Languagues from Positive Data, Information and Control 45, 117 - 135

[2] Angluin, D. and C.H. Smith, (1983), Inductive Inference: Theory and Methods, Computing Surveys 15, 3, 237 - 269

[3] Angluin, D. and C.H. Smith, (1987), Formal Inductive Inference, In Encyclopedia of Artificial Intelligence, St.C. Shapiro (Ed.), Vol. 1, pp. 409 - 418, Wiley-Interscience Publication, New York

[4] Blum, L. and M. Blum, (1975), Toward a Mathematical Theory of Inductive Inference, Information and Control 28, 122 - 155

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

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

[7] Case, J. and C. Lynes, (1982), Machine Inductive Inference and Language Identification, Proc. Automata, Languages and Programming, Ninth Colloquim, Aarhus, Denmark, M.Nielsen and E.M. Schmidt (Eds.), Lecture Notes in Computer Science 140, pp. 107 -115, Springer-Verlag

[8] Freivalds, R., Kinber, E. B. and R. Wiehagen, (1989), Inductive Inference from Good Examples, Proc. International Workshop on Analogical and Inductive Inference, October 1989, Reinhardsbrunn Castle, K.P. Jantke (Ed.), Lecture Notes in Artificial Intelligence 397, pp. 1 - 17, Springer-Verlag

[9] Freivalds, R., Kinber, E. B. and R. Wiehagen, (1992), Convergently versus Divergently Incorrect Hypotheses in Inductive Inference, GOSLER Report 02/92, January 1992, 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 A. Sharma, (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, Proc. First 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 543, pp. 161 - 177, Springer-Verlag

[15] Lange, S. and R. Wiehagen, (1991), Polynomial–Time Inference of Arbitrary Pattern Languages, New Generation Computing 8, 361 - 370

[16] Lange, S. and T. Zeugmann, (1991), Monotonic versus Non-monotonic Language Learning, in Proc. 2nd International Workshop on Nonmonotonic and Inductive Logic, December 1991, Reinhardsbrunn, to appear in Lecture Notes in Artificial Intelligence

[17] Lange, S. and T. Zeugmann, (1992A), On the Power of Monotonic Language Learning, GOSLER–Report 05/92, February 1992, Fachbereich Mathematik und Informatik, TH Leipzig

[18] Lange, S. and T. Zeugmann, (1992B), Types of Monotonic Language Learning and Their Characterizations, in Proc. 5th ACM Workshop on Computational Learning Theory, July 1992, Morgan Kaufmann Publishers Inc.

[19] Mukouchi, Y., (1991), Definite Inductive Inference as a Successful Identification Criterion, Research Institute of Fundamental Information Science, Kyushu University 33, Fukuoka, December 24, '91 RIFIS-TR-CS-52

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

[21] Solomonoff, R., (1964), A Formal Theory of Inductive Inference, Information and Control 7, 1 - 22, 234 - 254

[22] Trakhtenbrot, B.A. and Ya.M. Barzdin, (1970), Konetschnyje Awtomaty (Powedenie i Sintez), Nauka, Moskwa (in Russian)

[23] Wiehagen, R., (1976), Limes–Erkennung rekursiver Funktionen durch spezielle Strategien, J. Information Processing and Cybernetics (EIK) 12, 93 - 99

[24] Wiehagen, R., (1977), Identification of Formal Languages, Proc. Mathematical Foundations of Computer Science, Tatranska Lomnica, J. Gruska (Ed.), Lecture Notes in Computer Science 53, pp. 571 - 579, Springer-Verlag

[25] Wiehagen, R., (1991), A Thesis in Inductive Inference, in Proc. First International Workshop on Nonmonotonic and Inductive Logic, December 1990, Karlsruhe, J.Dix, K.P. Jantke and P.H. Schmitt (Eds.), Lecture Notes in Artificial Intelligence 543, pp. 184 - 207

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

[27] Zeugmann, T., (1983), A–posteriori Characterizations in Inductive Inference of Recursive Functions, Journal of Information Processing and Cybernetics (EIK) 19, 559 - 594