# Characterizations of Class Preserving Monotonic and Dual Monotonic Language Learning

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

Steffen Lange[1]
TH Leipzig
FB Mathematik und Informatik
PF 66
O–7030 Leipzig
steffen@informatik.th-leipzig.de

Shyam Kapur[2]
Institute for Research in Cognitive Science
University of Pennsylvania
3401 Walnut Street Rm 412C
Philadelphia, PA 19104, USA
skapur@linc.cis.upenn.edu

1992

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

[2] This work was supported in part by ARO grant DAAL 03-89-C-0031, DARPA grant N00014-90-J-1863, NSF grant IRI 90-16592 and Ben Franklin grant 91S.3078C-1.

## Abstract

The present paper deals with monotonic and dual monotonic language learning from positive as well as from positive and negative examples. The three notions of monotonicity reflect different formalizations of the requirement that the learner has to always produce 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 exclusively produce specializations that fit better and better to the target language. We characterize strong–monotonic, monotonic, weak–monotonic, dual strong–monotonic, dual monotonic and monotonic & dual monotonic as well as finite language learning from positive data in terms of recursively generable finite sets.

# 1. Introduction

The process of hypothesizing a general rule from eventually incomplete data is called inductive inference. Many philosophers of science have focused their attention on problems in inductive inference. Some of the principles developed are very much alive in *algorithmic learning theory*, a rapidly emerging science since the seminal papers of Solomonoff (1964) and of Gold (1967). The state of the art is excellently surveyed in Angluin and Smith (1983, 1987).

The present paper deals with formal language learning. In this field, many interesting and sometimes surprising results have been obtained within the last decades (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, or 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 correctly describing the object to be learnt. There are many possible requirements on the sequence of all actually created hypotheses. In all what follows we restrict ourselves to deal exclusively with *class preserving* language learning, i.e., we demand that all the hypotheses produced describe a language that is contained in the family of all target languages. This requirement seems to be a very natural one, since any hypothesis not fulfilling it cannot be correct. Furthermore, in the present paper, we mainly study language learning ¿from positive 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 (1991, 1992A, 1992B)). Subsequently, Kapur (1992B) 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 4 and 6, 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 4 (B)). That means, a new hypothesis is never allowed to reject some string that a previously generated guess already *correctly* includes.

On the other hand, it is only natural to consider the dual version of the latter requirement as well. Intuitively speaking, dual monotonicity describes the following requirement:

If, at any stage, the learner outputs a hypothesis *correctly* excluding a string $s$ from the language to be learnt, then any subsequent guess has to behave in the same way (cf. Definition 6 (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 4 and 6, part (C)).

In all what follows we restrict ourselves to deal exclusively with the learnability of indexed families of non–empty uniformly recursive languages. This case is of special interest with respect to potential applications. The first problem arising naturally is to relate 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, Zeugmann and Kapur (1992). 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 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 general setting of learning recursive functions the situation is totally different (cf. Freivalds, Kinber and Wiehagen (1992)). Looking at learning from positive data, the main problem consists in detecting or avoiding guesses that are supersets, i.e., overgeneralizations, of the language to be inferred. Obviously, conservative learners are never allowed to output an overgeneralized hypothesis. This restriction directly yields a limitation of learning power (cf. Angluin (1980), Lange, Zeugmann and Kapur (1992)). Moreover, Angluin (1980) proved a *characterization* theorem for inference from positive data that turned out to be very useful in applications. However, it remained open whether learning ¿from positive data that avoids overgeneralization may be characterized too. We solve this problem by characterizing almost all types of monotonic and dual monotonic language learning as well as of finite inference from positive data of recursive languages in terms of recursively generable families of recursive, non–empty and finite sets. As we shall see, the characterization theorems lead to a deeper insight into the problem of what may be inferred monotonically and dual monotonically. Moreover, in characterizing almost all types of monotonic and of dual monotonic inference, we provide a unifying framework to language learning. We shall discuss these issues throughout the paper in some more detail, since within the current level of precision nothing more satisfactory can be added.

Characterization theorems for monotonic as well as for dual monotonic language learning on informant can be found in Lange and Zeugmann (1992B).

The paper is structured as follows. Section 2 presents preliminaries, i.e., notations and definitions. The characterization theorems are established in Section 3 and the subsections therein. In Section 4, we discuss the results obtained and outline open problems. All references are given in Section 5.

## 2. Preliminaries

By $N = \{1, 2, 3, ...\}$ 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 $\Sigma$ we denote any fixed finite alphabet of symbols. Let $\Sigma^*$ be the free monoid over $\Sigma$. The length of a string $s \in \Sigma^*$ is denoted by $|s|$. Any subset $L \subseteq \Sigma^*$ is called a language. By $co - L$ we denote the complement of $L$, i.e., $co - L = \Sigma^* \setminus L$. Let $L$ be a language and $t = s_1, s_2, s_3, ...$ an infinite sequence of strings from $\Sigma^*$ 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 an infinite 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 $x \in N$. Then we set $t_x^+ = \{s_k \mid k \leq x\}$. Furthermore, by $i_x^+$ and $i_x^-$ we denote the sets $\{s_k \mid (s_k, +) \in i, k \leq x\}$ and $\{s_k \mid (s_k, -) \in i, k \leq x\}$, respectively.

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 $\Sigma$. Then $\mathcal{L}$ may be regarded as an indexed family of recursive languages (cf. Bucher and Maurer (1984)). In the sequel, we sometimes 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 depending on whether we want to synthesize in the limit grammars (i.e., procedures generating languages) or decision procedures, i.e., programs of characteristic functions. Case and Lynes (1982) investigated this phenomenon in detail. As it turns out, IIMs synthesizing grammars can be more powerful than those ones which are requested to output decision procedures. However, in the context of identification of indexed families, both concepts are of equal power. Nevertheless, we decided to require the IIMs to output grammars. This decision has been caused by the fact that there is a big difference between the possible monotonicity requirements. A straightforward adaptation of the approaches made in inductive inference of recursive functions directly yields analogous requirements with respect to the corresponding characteristic functions of the languages to be inferred. On the other hand, it is only natural to interpret monotonicity with respect to the language to be learnt, i.e., to require containment of languages as described in the introduction. As it turned out, the latter approach increases considerably the power of all types of monotonic and dual monotonic language learning. Furthermore, since we exclusively deal with *class preserving* learning of 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 $G_1, G_2, G_3, ...$ over the terminal alphabet $\Sigma$ 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 $s \in \Sigma^*$. 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 $(j_x)_{x\in N}$ 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 $\mathcal{G} = (G_j)_{j\in N}$ be a space of hypotheses. An IIM $M$ $LIM-TXT$ $(LIM-INF)$–identifies $L$ on a text $t$ (an informant $i$) with respect to $\mathcal{G}$ 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. Moreover, it may be well conceivable that the weakening of $\mathcal{L} = \{L(G_j) \mid j \in N\}$ to $\mathcal{L} \subseteq \{L(G_j) \mid j \in N\}$ may increase the collection of inferable indexed families. However, it does not, as the following proposition shows.

**Proposition 1. (Lange and Zeugmann, 1992C)** *Let $\mathcal{L}$ be an indexed family and let $\mathcal{G} = (G_j)_{j\in N}$ be any space of hypotheses such that $\mathcal{L} \subseteq \{L(G_j) \mid j \in N\}$ and membership in $L(G_j)$ is uniformly decidable. Then we have: If there is an IIM $M$ inferring $\mathcal{L}$ on* *text (informant) with respect to $\mathcal{G}$, then there is also an IIM $\hat{M}$ that learns $\mathcal{L}$ on text (informant) with respect to $\mathcal{L}$.*

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 $\mathcal{G} = (G_j)_{j \in N}$ be a space of hypotheses. An IIM $M$ $FIN - TXT \quad (FIN - INF)$–identifies $L$ on a text $t$ (on an informant $i$) with respect to $\mathcal{G}$ iff it outputs only a single and correct hypothesis $j$, i.e., $L = L(G_j)$, and stops thereafter.*

*Moreover, $M$ $FIN-TXT \quad (FIN-INF)$–identifies $L$, iff $M$ $FIN-TXT \quad (FIN-INF)$–identifies $L$ on every text (informant) for $L$. We set: $FIN - TXT(M) = \{L \in \mathcal{L} \mid M\ FIN - TXT - \text{identifies } L\}$, and define $FIN - INF(M)$ analogously.*

The resulting identification type is denoted by $FIN - TXT \quad (FIN - INF)$.

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

**Definition 3. (Barzdin, 1974)** *An IIM $M$ $CONS - TXT \quad (CONS - INF)$–identifies $L$ on a text $t$ (an informant $i$) iff*

*(1) $M$ $LIM - TXT \quad (LIM - INF)$–identifies $L$ on $t$ (on $i$)*

*(2) Whenever $M$ on $t_x$ ($i_x$) produces a hypothesis $j_x$, then $t_x^+ \subseteq L(G_{j_x})$ ($i_x^+ \subseteq L(G_{j_x})$ and $i_x^- \subseteq co - L(G_{j_x})$).*

*$M$ $CONS-TXT$ $(CONS-INF)$–identifies $L$ iff $M$ $CONS-TXT$ $(CONS-INF)$–identifies $L$ on every text $t$ (informant $i$).*
*By $CONS - TXT(M) \quad (CONS - INF(M))$ we denote the set of all languages which $M$ $CONS - TXT \quad (CONS - INF)$–identifies. $CONS - TXT$ and $CONS - INF$ are analogously defined as above.*

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

**Definition 4. (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 $L(G_{j_x}) \subseteq L(G_{j_{x+k}})$).*

Remark: (C) in particular means that $M$ has to work strong–monotonically as long as its guess $j_x$ is consistent with the data fed to $M$ after $M$ has output $j_x$.

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 languages (cf. e.g. Lange and Zeugmann (1991, 1992A)).

Figure 1 summarizes the known results concerning monotonic language learning (cf. Lange and Zeugmann (1991)). Incomparability of sets is denoted by #.

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

**Figure 1**

Next to, we define *conservatively* working IIMs.

**Definition 5. (Angluin, 1980)**
*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)* $L \in LIM - TXT(M)$ $(L \in LIM - INF(M))$

*(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 $s \in i^+_{x+k}$ or it generates some string $s \in i^-_{x+k}$).*

*CONSERVATIVE–TXT*$(M)$ and *CONSERVATIVE –INF(M)* as well as the collections of sets *CONSERVATIVE–TXT* and *CONSERVATIVE–INF* are defined in a manner analogous to that 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*.

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

**Definition 6.** *An IIM $M$ is said to identify a language $L$ from text (informant)*

*(A) dual strong–monotonically*

*(B) dual monotonically*

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

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

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

By $SMON^d-TXT$, $SMON^d-INF$, $MON^d-TXT$, $MON^d-INF$, $WMON^d-TXT$, and $WMON^d-INF$ we denote the collections of all those indexed families $\mathcal{L}$ of languages for which there is an IIM identifying it dual strong–monotonically, dual monotonically and dual weak–monotonically from text and informant, respectively.

The next figure shows the relations between the defined modes of dual monotonic inference (cf. Lange, Zeugmann and Kapur (1992)). Compared with Figure 1 it may help to illustrate the similarities as well as the differences between the types of monotonic and of dual monotonic inference.

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

**Figure 2**

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

Next, we combine the monotonicity constraints ¿from Definition 4 and Definition 6. As we shall see, this might help to gain a better understanding of the relationships between monotonic inference of languages and other well–known types of language learning.

**Definition 7.** *Let* $SMON^{\&} - TXT$ *(*$SMON^{\&} - INF$*) denote the class of indexed families learnable by an IIM that works strong–monotonically as well as dual strong–monotonically. The identification types* $MON^{\&} - TXT$*,* $MON^{\&} - INF$*,* $WMON^{\&} - TXT$ *and* $WMON^{\&} - INF$ *are defined analogously.*

Finally, we present two figures relating all types of monotonic language learning to all types of dual monotonic inference of recursive languages. For the sake of readability, we separate the results for language learning on text from those ones dealing with language learning ¿from positive and negative data.

The lines between the identification types indicate set inclusion, i.e., the lower type is properly contained in the upper one. Missing lines indicate incomparability of the collections of sets.

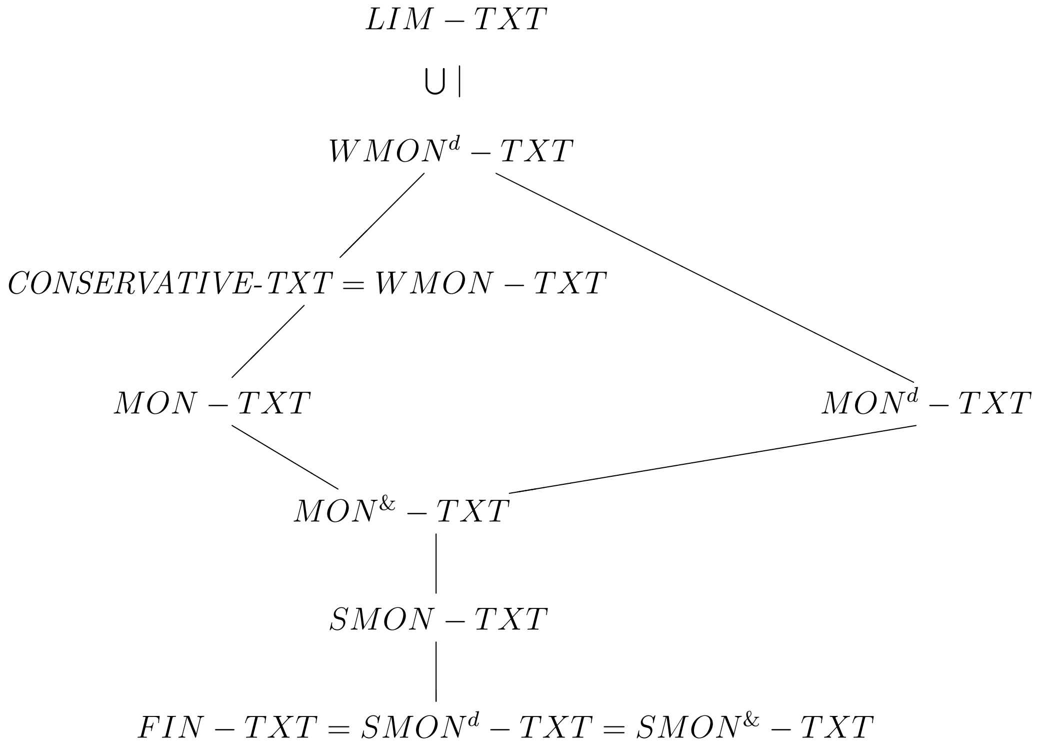

**Figure 3**

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

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

**Figure 4**

Additional results which give further insights into the relations between the defined identification types can be found in Lange, Zeugmann and Kapur (1992).

## 3. Characterization Theorems

In this section, we present characterizations of all types of monotonic and of dual monotonic language learning from positive data. Characterizations play an important role in that they lead to 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 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. Furthermore, characterizations may help gain a better understanding of the properties objects should have in order to be inferable in the desired sense. A very illustrative example is Angluin's (1980) characterization of those indexed families for which learning in the limit from positive data is possible. In particular, this theorem provides insight into the problem how to deal with overgeneralizations. Theorem 5 below offers an alternative way to resolve this question.

**Proposition 2. (Angluin, 1980)** *Let $\mathcal{L}$ be an indexed family of recursive languages. Then: $\mathcal{L} \in LIM-TXT$ if and only if there is an effective procedure which on any input $j \in N$ enumerates a tell–tale set $T_j$ of strings such that*

*(1) $T_j$ is finite.*

*(2) For all $j \in N$, $T_j \subseteq L_j$.*

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

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

Though Angluin (1980) established some sufficient conditions that guarantee conservative learning from positive data, it remained open whether *CONSERVATIVE–TXT* may be characterized in terms of finite non–empty sets. So let us start this section with a solution to this long standing open problem. The characterization of *CONSERVATIVE–TXT* has been obtained by developing two new ideas. First, looking at Proposition 1 and 2, it might seem that the particular choice of the space of hypotheses is negligible. However, when dealing with conservative learning, the situation is totally different (cf. Lange, Zeugmann and Kapur (1992)). Therefore, in characterizing conservative learning one has to construct an appropriate space of hypotheses. Second, this construction is combined with an effective procedure generating recursive tell–tale sets rather than recursively enumerable ones as in Proposition 2. Finally, we state the announced theorem as a characterization of weak–monotonic language learning from positive data.

## 3.1. The Characterization of Weak–Monotonic Inference

Our first theorem characterizes $WMON-TXT$ in terms of recursively generable finite tell–tales. A family of finite sets $(T_j)_{j\in N}$ is said to be recursively generable iff there is a total effective procedure $g$ which, on input $j$, generates all elements of $T_j$ and stops. If the computation of $g(j)$ stops and there is no output, then $T_j$ is considered to be empty. Finally, for notational convenience, we use $L(\mathcal{G})$ to denote $\{L(G_j) \mid j \in N\}$ 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: $\mathcal{L} \in WMON-TXT$ 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) 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* $L(\hat{G}_z) \not\subset L(\hat{G}_j)$.

*Proof.* Necessity: Let $\mathcal{L} \in WMON{-}TXT = CONSERVATIVE{-}TXT$. Then there are an IIM $M$ and a space of hypotheses $(G_j)_{j\in N}$ such that $M$ infers any $L \in \mathcal{L}$ conservatively with respect to $(G_j)_{j\in N}$. We proceed in showing how to construct $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in N}$. This is done in two steps. First, we construct a space of hypotheses $\tilde{\mathcal{G}} = (\tilde{G}_j)_{j\in N}$ as well as a recursively generable family $(\tilde{T}_j)_{j\in N}$ of finite but possibly empty sets. Then, we describe a procedure enumerating a certain subset of $\tilde{\mathcal{G}}$ which we call $\hat{\mathcal{G}}$. Let $c : N \times N \to N$ be Cantor's pairing function. We define the space of hypotheses $(\tilde{G}_j)_{j\in N}$ as well as the wanted family $(\tilde{T}_j)_{j\in N}$ as follows: On input $j$ compute $k, x \in N$ such that $j = c(k,x)$. Then set $\tilde{G}_{c(k,x)} = G_k$. Furthermore, for any language $L(G_k)$, we denote by $t^k$ the canonically ordered text of $L(G_k)$ defined as follows: Let $s_1, s_2, ...$ be the lexicographically ordered text of $\Sigma^*$. Test sequentially whether $s_z \in L(G_k)$, for $z = 1, 2, 3, ...$, until the first $z$ is found such that $s_z \in L(G_k)$. Since $L(G_k) \neq \emptyset$, there must be at least one $z$ fulfilling the test. Set $t_1^k = s_z$. We proceed inductively:

$$t_{x+1}^k = \begin{cases} t_x^k s_{z+x+1} & if \quad s_{z+x+1} \in L(G_k) \\ t_x^k s & otherwise,\ where\ s \\ & is\ the\ last\ string\ in\ t_x^k \end{cases}$$

We define:

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

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

for any $x \geq y$. This proves property (1). Finally, we have to show (3). It results from the requirement that any conservatively working IIM is never allowed to output an overgeneralized hypothesis, i.e., a guess that generates a proper superset of the language to be inferred. To see this, suppose the converse, i.e., there are $j,\ z \in N$ such that $\hat{T}_j \subseteq L(\hat{G}_z)$ and $L(\hat{G}_z) \subset L(\hat{G}_j)$. By definition, there are uniquely determined $k, x \in N$ such that $j = c(k, x)$. Let $s_1, ..., s_y$ be the strings of $\hat{T}_j$ in canonical order with respect to $L(\hat{G}_{c(k,x)})$. By construction we obtain $M(s_1, ..., s_y) = k$. Now we conclude that $s_1, ..., s_y$ is an initial segment of the canonically ordered text for $L(\hat{G}_z)$, since $\hat{T}_j \subseteq L(\hat{G}_z) \subset L(\hat{G}_j) = L(\hat{G}_{c(k,x)})$. Finally, $M$ has to infer $L(\hat{G}_z)$ on its canonically ordered text. Thus, it has to perform a mind change in some subsequent step which cannot be caused by an inconsistency. This contradiction yields (3).

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

$M(t_x) =$ "If $x = 1$ or $M$ on $t_{x-1}$ does not output a hypothesis, then goto (B). Otherwise, goto (A).

(A) Let $j$ be the hypothesis produced last by $M$ when fed with $t_{x-1}$. Test whether $t_x^+ \subseteq L(\hat{G}_j)$. In case it is, output $j$ and request the next input. Otherwise, goto (B).

(B) For $j = 1, ..., x$, generate $\hat{T}_j$ and test whether $\hat{T}_j \subseteq t_x^+ \subseteq L(\hat{G}_j)$. In case there is at least a $j$ fulfilling the test, output the minimal one. Otherwise, output nothing and request the next input."

Since all of the $\hat{T}_j$ are uniformly recursively generable and finite, we see that $M$ is an IIM. Now it suffices to show that $M$ infers $L$ on $t$ conservatively. Since the machine $M$ changes its mind only in case it finds an inconsistency in (A), it works conservatively.

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

Let $z = \mu k[L = L(\hat{G}_k)]$. Consider $\hat{T}_1,\ ...,\ \hat{T}_z$. Then there must be an $x$ such that $\hat{T}_z \subseteq t_x^+ \subseteq L(\hat{G}_z)$. That means, at least after having fed $t_x$ to $M$, the machine $M$ outputs a hypothesis. Furthermore, after having fed $t_x$ to $M$, the machine $M$ always outputs a hypothesis and it never outputs a guess $j > x$ since $z \in \{k \leq x \mid \hat{T}_k \subseteq t_x^+ \subseteq L(\hat{G}_k)\}$. Moreover, since $M$ changes its mind if and only if it receives some text string that misclassifies its current guess, we see that any rejected hypothesis is never repeated in some subsequent step. Finally, since at least $z$ can never be rejected, $M$ has to converge.

*Claim* 2. If $M$ converges, say to $j$, then $L = L(\hat{G}_j)$.

Suppose the converse, i.e., $M$ converges to $j$ and $L \neq L(\hat{G}_j)$.

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

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

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

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

q.e.d.

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

Kapur and Bilardi (1992) also established characterizations of conservative learning. Their main characterization differs at least conceptually from the one presented above. In order to see the difference, we need the following notion. Let $A$ be a finite set and let $\mathcal{L}$ be an indexed family of languages. A language $L \in \mathcal{L}$ is said to be a least upper bound of $A$ iff $A \subseteq L$ and any language $\hat{L} \in \mathcal{L}$ containing $A$ is not a proper subset of $L$. Kapur and Bilardi (1992) showed that conservative learning is equivalent to the existence of a recursive enumeration of pairs of finite sets and grammars such that, in each pair, the language corresponding to the grammar is a least upper bound of the corresponding finite set, and, for each $L \in \mathcal{L}$, there is at least a corresponding pair. Consequently, this characterization is conceptually based on the judicious use of a function computing least upper bounds. Our approach is in some sense converse in that we construct a suitable enumeration $\hat{\mathcal{L}}$ of $\mathcal{L}$ and for every language $\hat{L} \in \hat{\mathcal{L}}$ a recursive and finite set such that $\hat{L}$ is a least upper bound of it.

## 3.2. The Characterization of Strong–Monotonic Inference

Next we characterize $SMON - TXT$. As it turned out, the same proof technique presented above applies mutatis mutandis to obtain the following theorem.

**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) 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 $L(\hat{G}_j) \subseteq L(\hat{G}_z)$.*

*Proof.* Necessity: Let $\mathcal{L} \in SMON - TXT(M)$. The recursively generable family $(\hat{T}_j)_{j \in N}$ of finite and non–empty sets is analogously defined as in in the proof of Theorem 1. Using the same arguments as above one, immediately obtains property (1) and (2). In order to prove (3), let $\hat{T}_j \subseteq L(\hat{G}_z)$. We have to show that $L(\hat{G}_j) \subseteq L(\hat{G}_z)$. Let $k,\ x$ be the uniquely determined numbers with $j = c(k, x)$. Furthermore, let $s_1, ..., s_y$ be the strings of $\hat{T}_j$ in canonical order with respect to $L(\hat{G}_{c(k,x)})$ such that $M(s_1, ..., s_y) = k$ for the first time. Since $\hat{T}_j \subseteq L(\hat{G}_z)$, we see that $s_1, ..., s_y$ is also an initial segment of some text for $L(\hat{G}_z)$. Consequently, $s_1, ..., s_y$ may be extended to a text for $L(\hat{G}_z)$ . Finally, since $M$ has to infer $L(\hat{G}_z)$ too, there should be an $n \in N$ and a finite extension $\sigma$ of strings of $L(\hat{G}_z)$ such that $M(s_1, ..., s_y, \sigma) = n$ and $L(\hat{G}_z) = L(G_n)$. $M$ works strong–monotonically and hence, by the transitivity of $\subseteq$, we obtain $L(\hat{G}_j) \subseteq L(\hat{G}_z)$.

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

$M(t_x) =$ "Generate $\hat{T}_j$ and test whether $\hat{T}_j \subseteq t_x^+ \subseteq L(\hat{G}_j)$ for $j = 1, ..., x$. 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."

We have to show that $M$ infers $\mathcal{L}$ strong–monotonically. Since all of the $\hat{T}_j$ are uniformly recursively generable and finite we see that $M$ is an IIM. First we show that $M$ identifies $L$ on $t$. Let $k = \mu z[L = L(\hat{G}_z)]$. We claim that $M$ converges to $k$. Consider $\hat{T}_1, ..., \hat{T}_k$. Then there must be an $x$ such that $\hat{T}_k \subseteq t_x^+ \subseteq L(\hat{G}_k)$. Thus, at least after having fed $t_x$ to $M$ the machine must output a guess. Moreover, since for all $r \in N$ we additionally have $\hat{T}_k \subseteq t_{x+r}^+ \subseteq L(\hat{G}_k)$, we may conclude that after having fed $t_x$ to $M$, it never produces a hypothesis $j > k$. Suppose $M$ converges to $j < k$. Due to the choice of $k$ we know $L(\hat{G}_j) \neq L(\hat{G}_k) = L$.
*Case* 1. $L \subset L(\hat{G}_j)$

By construction, if $M$ outputs $j$ at all, then there should be an $n \in N$ such that $\hat{T}_j \subseteq t_n^+ \subseteq L(\hat{G}_j)$. Moreover, since $t$ is a text for $L = L(\hat{G}_k)$, we know that $t_n^+ \subseteq L(\hat{G}_k)$ for all $n \in N$. Hence $\hat{T}_j \subseteq L(\hat{G}_k)$. Now we can apply property (3) and obtain $L(\hat{G}_j) \subseteq L(\hat{G}_k) = L$, a contradiction. Moreover, a closer look at the latter argument shows that $M$ can never output an overgeneralized hypothesis.
*Case* 2. $L \setminus L(\hat{G}_j) \neq \emptyset$
Again, suppose that $M$ converges to $j < k$. Let $s \in L \setminus L(\hat{G}_j)$. Thus, there must be an $n \in N$ such that $s \in t_n^+$. Consequently, after having seen at least $t_n^+$, the machine $M$ cannot output $j$.

Summarizing, we obtain that $M$ converges to $k$. It remains to show that $M$ works strong–monotonically. Suppose, $M$ outputs $y$ and changes its mind to $z$ in some subsequent step. By construction we have: $\hat{T}_y \subseteq t_n^+ \subseteq L(\hat{G}_y)$, for some $n \in N$, and $\hat{T}_z \subseteq t_{n+r}^+ \subseteq \hat{T}_z$, for some $r > 0$. But now, $\hat{T}_y \subseteq t_{n+r}^+ \subseteq L(\hat{G}_z)$, and again we conclude ¿from property (3) that $L(\hat{G}_y) \subseteq L(\hat{G}_z)$. Hence, $M$ indeed works strong–monotonically on $t$.

q.e.d.

The characterization theorem of $SMON - TXT$ has the following interesting 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 to solve the open problem whether or not set inclusion of pattern languages is decidable.

## 3.3. The Characterization of Dual Strong–Monotonic Inference

In this subsection, we present the characterization of $SMON^d - TXT$ and postpone that one for $MON - TXT$ for a moment, since it deserves special attention. Since $SMON^d - TXT = SMON^{\&} - TXT = FIN - TXT$, it suffices to characterize $FIN - TXT$ in order to obtain the intended characterization of dual strong–monotonic inference as well as of $SMON^{\&} - TXT$. Note that a bit weaker theorem has been obtained independently by Mukouchi (1991). The difference is caused by Mukouchi's definition of finite identification from text, since there it is demanded that any indexed family $\mathcal{L}$ has to be finitely inferred with respect to $\mathcal{L}$ itself. Consequently, one should ask whether or not the latter requirement might lead to a decrease in the inferring power. It does not, as we shall see.

Our next characterization has some special features distinguishing it ¿from the characterizations already given. As pointed out above, dealing with characterizations has been motivated by the aim to elaborate a unifying approach to monotonic inference. Concerning $WMON - TXT$ as well as $SMON - TXT$ this goal has been completely met by showing that there is essentially one algorithm, i.e., that one described in the proof of Theorem 1 and Theorem 2, respectively, which can perform the desired inference task, if the space of hypotheses is appropriately chosen. The next theorem yields an even stronger implication. Namely, it shows that, if there is a space of hypotheses at all such that $\mathcal{L} \in FIN - TXT$ 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 3.** *Let $\mathcal{L}$ be an indexed family of recursive languages. Then: $\mathcal{L} \in FIN - TXT$ if and only if there is a recursively generable family $(T_j)_{j \in N}$ of finite non–empty sets such that*

*(1) $T_j \subseteq L_j$ for all $j \in N$.*

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

*Proof.* Necessity: Let $\mathcal{L} \in FIN - TXT$. Then, there are a space $\mathcal{G} = (G_j)_{j \in N}$ 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 $(T_j)_{j \in N}$. This is done in two steps. First, we construct $(\hat{T}_j)_{j \in N}$

with respect to the space $\mathcal{G}$ of hypotheses. Then, we describe a procedure yielding the wanted family $(T_j)_{j\in N}$ with respect to $\mathcal{L}$.

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

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

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

$M(t_x) =$ "Generate $T_j$ for $j = 1, ..., x$ and test whether $T_j \subseteq t_x^+ \subseteq L_j$.

In case there 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 $T_j$ 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 $T_j \subseteq t_x^+$. That means, at least after having fed $t_x$ to $M$, the machine $M$ outputs a hypothesis and stops. Suppose $M$ produces a hypothesis $k$ with $k \neq j$ and stops. Hence, there has to be a $z$ with $z < x$ such that $T_k \subseteq t_z^+$. Since $z < x$, it follows $T_k \subseteq L_j$. Thus, (2) implies $L_k = L_j$. Consequently, $M$ outputs a correct hypothesis for $L$ and stops afterwards.

q.e.d.

## 3.4. The Characterization of Monotonic Inference

Next, we characterize $MON - TXT$. As it turned out, characterizing $MON - TXT$ is much more complicated. Intuitively this is caused by the following observations. 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 segment of the text 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. Nevertheless, at first glance there might be some hope. Osherson, Stob and Weinstein (1986) defined *set–driven* as well as *rearrangement–independent* IIMs. An IIM $M$ is set–driven (rearrangement–independent) iff its output depends only on the range of its input (only on the range and length of its input). However, set–driveness is a very restrictive requirement (cf. Osherson et al. (1986), Fulk (1990)). On the other hand, Fulk (1990) proved that any IIM $M$ may be replaced by an IIM $M'$ which is rearrangement–independent. Unfortunately, $M'$ does not preserve any of the types of monotonicity. Nevertheless, strong–monotonic inference may always be performed by an IIM working rearrangement–independent as the proof of Theorem 2 shows. Surprisingly enough, the IIM described in the proof of Theorem 1 is not rearrangement–independent. But it possesses another favorable property, i.e., the hypothesis it converges to is the first correct one in the sequence of all created guesses. IIMs fulfilling this property are said to work *semantically finite.* While the IIM described in the demonstration of Theorem 2 does not necessarily work semantically finite, it may, however, be replaced by an IIM $M'$ that works *strong–monotonically, rearrangement–independent* and *semantically finite.* In fact, $M'$ works exactly as $M$ does but it uses a different space of hypotheses. A closer look at property (3) yields the following interesting consequence. If $\mathcal{L} \in SMON-TXT$, then there is a recursive enumeration of $\mathcal{L}$, i.e., that one constructed in the proof, such that for any $k,\ j \in N$ it is uniformly decidable whether or not $L(\hat{G}_k) \subseteq L(\hat{G}_j)$. Hence, equality of languages is uniformly decidable. Thus, one may construct the wanted space of hypotheses containing each language of $\mathcal{L}$ exactly once.

On the other hand, it remained open whether the IIM presented in the proof of Theorem 1 may be replaced by an IIM that works semantically finite and rearrangement–independent. We conjecture it cannot. So it seems that rearrangement–independence gets lost somewhere in the hierarchy of monotonic inference. We conjecture that monotonic inference from positive data performed by rearrangement–independent IIMs is less powerful than ordinary monotonic learning from text. Summarizing the discussion above, in characterizing $MON - TXT$ we have to overcome the difficulties pointed out in a different way. Wiehagen (1992) proposed to construct for every language $L$ and for *every* text $t$ of $L$ a family of characteristic finite sets and obtained a characterization theorem that is close to his characterization of monotonic inference of total recursive functions (cf. Wiehagen (1991)). However, conceptually it differs completely from the theorems presented above.

What we now present is a characterization of $MON-TXT$ in terms of recursively generable finite sets as above. Additionally, we have been forced to define an easy computable relation $\prec \subseteq N \times N$ that can be used to distinguish appropriate chains of tell–tales with the help of which an IIM $M$ may compute its hypotheses. Now we are ready to present the wanted characterization.

**Theorem 4.** *Let $\mathcal{L}$ be an indexed family of recursive languages. Then: $\mathcal{L} \in MON-TXT$ if and only if there is a space of hypotheses $\hat{\mathcal{G}} = (\hat{G}_j)_{j \in N}$, a computable relation $\prec$ over $N$, 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) For all $L \in \mathcal{L}$ and all $k \in N$,*

*(i) $\hat{T}_k \subseteq L(\hat{G}_k)$.*

*(ii) if $\hat{T}_k \subseteq L$, then $L \not\subset L(\hat{G}_k)$.*

*(3) For all $L \in \mathcal{L}$ and any $k \in N$, and all finite $A \subseteq L$, if $\hat{T}_k \subseteq L$, $L(\hat{G}_k) \neq L$, then there is a $j$ such that $k \prec j$, and $A \subset \hat{T}_j \subseteq L(\hat{G}_j) = L$.*

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

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

*Proof.* Necessity: We start by defining the relation $\prec$. For this purpose some additional notation is needed. Let $N^*$ be the set of all finite sequences over $N$, and for $\alpha \in N^*$ let $|\alpha|$ denote the length of $\alpha$. Whenever appropriate, we interpret a number $k$ as a bijective encoding of a 4-tupel $(n, \alpha, \beta, \gamma)$, where $n \in N$, $\alpha$, $\beta$, $\gamma \in N^*$. Let $k,\ j \in N$. Then $k \preceq j$ iff $k = (n, \alpha, \beta, \gamma)$, $j = (m, \alpha\delta, \beta n \tau, \gamma\kappa)$, where $|\alpha| = |\beta| = |\gamma|$ as well as $|\delta| = |n\tau| = |\kappa|$. Moreover, $k \approx j$ iff $k = (n, \alpha, \beta, \gamma)$, $j = (m, \alpha\delta, \beta n \tau, , \gamma\kappa)$, where $|\alpha| = |\beta| = |\gamma|$ as well as $|\delta| = |n\tau| = |\kappa|$ and $range(\tau) = \{n\}$. Finally, $k \prec j$ iff $k \preceq j$ and not $k \approx j$. Note that $\prec$ is a transitive relation.

Let $M$ be an IIM inferring $\mathcal{L}$ without loss of generality monotonically, conservatively and consistently with respect to some space $\mathcal{G} = (G_j)_{j \in N}$ of hypotheses (cf. Lange and Zeugmann (1991)). Furthermore, for technical convenience, $M$ initially always outputs 0, where $L_0 = \emptyset$. For all $n \in N$, $\alpha, \beta, \gamma \in N^*$ we define $\tilde{G}_{(n,\alpha,\beta,\gamma)} = G_n$, and set $G_0 = \emptyset$. Moreover, we define $\tilde{T}_{(n,\alpha,\beta,\gamma)}$ as follows:

(i) If not $(|\alpha| = |\beta| = |\gamma|)$, then $\tilde{T}_{(n,\alpha,\beta,\gamma)} = \emptyset$.

(ii) If $(|\alpha| = |\beta| = |\gamma|)$, $\beta = \hat{\beta} n \rho$, $n \notin range(\hat{\beta})$, and $range(\rho) \neq \{n\}$, then $\tilde{T}_{(n,\alpha,\beta,\gamma)} = \emptyset$.

(iii) Otherwise let $\alpha = y_1, ..., y_r$, $\beta = 0, n_1, ..., n_{r-1}$, $\gamma = z_1, ..., z_r$ and do the following:

Generate the canonical text $\hat{\sigma}_{n_1}$ of $L(G_{n_1})$ of length $y_1$ and compute $visible(\hat{\sigma}_{n_1}) = \{\tau \mid |\tau| \leq |\hat{\sigma}_{n_1}|,\ range(\tau) \subseteq range(\sigma_{n_1})\}$. If $|visible(\hat{\sigma}_{n_1})| < z_1$, then $\tilde{T}_{(n,\alpha,\beta,\gamma)} = \emptyset$. Else test whether the $z_1$st element $\sigma_{n_1}$ (with respect to the

lexicographical ordering) of $visible(\hat{\sigma}_{n_1})$ when fed successively to $M$ exactly yields the sequence $0, n_1$ of hypotheses. If not, then $\tilde{T}_{(n,\alpha,\beta,\gamma)} = \emptyset$, and stop.
In case it does, generate the canonical text $\hat{\sigma}_{n2}$ of $L(G_{n2})$ of length $y_1 + y_2$. Compute $visible(\hat{\sigma}_{n_2})$, and test whether $|visible(\hat{\sigma}_{n_2})| < z_2$. In case it is, set $\tilde{T}_{(n,\alpha,\beta,\gamma)} = \emptyset$ and stop. For if not, test whether the $z_2$nd element of $\sigma_{n_2}$ (with respect to the lexicographical ordering) of $visible(\hat{\sigma}_{n_2})$ does fulfil the following properties:

(a) $\sigma_{n_1}$ is a prefix of $\sigma_{n_2}$ , and

(b) When fed successively with $\sigma_{n_2}$, the machine $M$ exactly produces the sequence $0, n_1, n_2$ of guesses.

In case it does not, set $\tilde{T}_{(n,\alpha,\beta,\gamma)} = \emptyset$ and stop. Otherwise continue analogously. Finally, if $\tilde{T}_{(n,\alpha,\beta,\gamma)}$ has not been defined til now, generate the canonical text $\hat{\sigma}_n$ of $L(G_n)$ of length $y_1 + ... + y_r$ and compute $visible(\hat{\sigma}_n)$. If $|visible(\hat{\sigma}_n)| < z_r$, then set $\tilde{T}_{(n,\alpha,\beta,\gamma)} = \emptyset$. Else let $\sigma_n$ be the $z_r$th element of $visible(\hat{\sigma}_n)$ with respect to the lexicographical ordering. Test whether

(a) $\sigma_{n_{r-1}}$ is a prefix of $\sigma_n$

(b) When fed successively with $\sigma_n$, the machine $M$ exactly produces the sequence $0, n_1, n_2, ..., n_{r-1}, n$ of guesses.

In case the test is not completely fulfilled set $\tilde{T}_{(n,\alpha,\beta,\gamma)} = \emptyset$. Otherwise, set $\tilde{T}_{(n,\alpha,\beta,\gamma)} = range(\sigma_n)$.

Obviously, $\tilde{T}_{(n,\alpha,\beta,\gamma)}$ is uniformly recursively generable and finite. The families $(\hat{T}_j)_{j\in N}$ as well as $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in N}$ are again obtained by simply striking off all $\tilde{T}_j$ that are empty as well as the corresponding $\tilde{G}_j$.

It remains to show that property (1) to (5) are satisfied. In order to prove (1), let $L \in \mathcal{L}$, and let $n \in N$ be chosen such that $M$ on $L$'s canonical text $t^L$ converges to $n$. Furthermore, let $y = \mu z[M(t^L_z) = n]$. We proceed in showing that there are $\alpha, \beta, \gamma \in N^*$ such that $\tilde{T}_{(n,\alpha,\beta,\gamma)} \neq \emptyset$. The latter statement yields (1), since we may conclude that $L(\hat{G}_{(n,\alpha,\beta,\gamma)}) = L(G_n) = L$. We define $\beta$ to be the sequence $0, n_1, n_2, ..., n_r$ of $M$'s hypotheses when successively fed with $t^L_y$ where the last element $n$ is deleted.
Let $\sigma_{n_1} \sqsubset \sigma_{n_2} \sqsubset ... \sqsubset \sigma_{n_r} \sqsubset t^L_y$ be the corresponding initial segments of text on which $M$ produces its hypotheses (* $\sqsubset$ denotes prefix relation of finite sequences *). Since $M$ works consistently, we obtain $\sigma^+_{n_j} \subseteq L(G_{n_j})$ for $j = 1, ..., r$. Compute the canonical text $t^{n_1}_{a_1}$, $t^{n_1}_{a_2}, ...$ of $L(G_{n_1})$ of length $a_i = |\sigma_{n_1}| + i$, $i = 0, 1, ...$ until the least $i$ with $\sigma_{n_1} \in visible(t^{n_1}_{a_i})$ has been found. Then let $z_1$ be the lexicographical number of $\sigma_{n_1}$ with respect to $visible(t^{n_1}_{a_i})$, and set $y_1 = a_i$. Next generate the canonical text $t^{n_2}_{a_1}$, $t^{n_2}_{a_2}, ...$ of $L(G_{n_2})$ of length $a_i = |\sigma_{n_2}| + i$ until $\sigma_{n_2} \in visible(t^{n_2}_{a_i})$ for the first time. Then define $z_2$ to be the lexicographical number of $\sigma_{n_2}$ with respect to $visible(t^{n_2}_{a_i})$, and set $y_2 = a_i - y_1$, a.s.o.

Finally, suppose $y_1, ..., y_r$ as well as $z_1, ..., z_r$ are already defined. Then we set $y_{r+1} = max\{y,\ y_1 + ... + y_r\}$. Define $z_{r+1}$ to be lexicographical number of $t^L_y$ with respect to

$visible(t^L_{y_{r+1}})$. Then, setting $\alpha = y_1, ..., y_{r+1}$, $\beta = 0, n_1, ..., n_r$ and $\gamma = z_1, ..., z_{r+1}$ directly yields $\tilde{T}_{(n,\alpha,\beta,\gamma)} \neq \emptyset$. This proves part (1).

Assertion (i) of (2) follows directly from the construction described above. The second part of property (2) is an immediate consequence of the conservativeness of $M$ (cf. proof of Theorem 1).

The technique applied above to prove (1) applies mutatis mutandis to obtain (3). Hence, we describe only the modification that has to be made. Let $A \subseteq L$ and let $\sigma = s_1, ..., s_m$ be the sequence of $A$'s strings written in lexicographical order. Moreover, let $\hat{T}_k \subseteq L$ and $L(\hat{G}_k) \neq L$. Then there are $n \in N$, $\alpha, \beta, \gamma \in N^*$ with $|\alpha| = |\beta| = |\gamma|$ such that $k = (n, \alpha, \beta, \gamma)$. Furthermore, let $\alpha = y_1, ..., y_r$, $\beta = 0, n_1, ..., n_{r-1}$, $\gamma = z_1, ..., z_r$, $q = y_1 + ... + y_r$, and let $w_1, ..., w_q$ be the uniquely determined sequence of all elements of $\hat{T}_k$ on which, when fed successively to $M$, the machine $M$ produces $\beta$ as its sequence of hypotheses. Since $\hat{T}_k \subseteq L$ but $L \neq L(\hat{G}_k) = L(\hat{G}_{(n,\alpha,\beta,\gamma)})$, we conclude that $M$ has not yet converged on this particular initial segment $w_1, ..., w_q$ of some text for $L$. Next we consider $M$'s behavior when fed with $w_1, ..., w_q, \sigma$. There are two cases to distinguish, i.e., either the computation of $M(w_1, ..., w_q, \sigma)$ ends in $M$'s request state, or it yields a guess. However, in both cases we extend $w_1, ..., w_q, \sigma$ with a sufficiently long initial segment $t^L_y$ of $L$'s canonical text until $M$ outputs a hypothesis that is correct for $L$. Finally, $j$ is obtained analogously as in the proof of property (1) where the construction is performed with respect to $w_1, ..., w_q, \sigma, t^L_y$ instead of $t^L_y$. Obviously, $k \prec j$, $A \subseteq \hat{T}_j$ and $L(\hat{G}_j) = L$. Hence (3) is proved.

We continue in proving property (4). Recall the definition of the relation $\prec$. Since $k \prec j$, in particular there are $n, m \in N$, $\alpha, \beta, \gamma, \delta, \tau, \kappa \in N^*$ such that $k = (n, \alpha, \beta, \gamma)$ and $j = (m, \alpha\delta, \beta n\tau, \gamma\kappa)$. Due to the definition of $\hat{T}_j$, we obtain an initial segment $\sigma$ of text for $L$ on which $M$, when successively fed with, sometime outputs $n$, and in some subsequent step $m$. Taking into account that $M$ works monotonically we obtain $L(G_n) \cap L \subseteq L(G_m) \cap L$. Finally, in accordance with our construction we know that $L(\hat{G}_k) = L(\hat{G}_{(n,\alpha,\beta,\gamma)}) = L(G_n)$ and $L(\hat{G}_j) = L(\hat{G}_{(m,\alpha\delta,\beta n\tau,\gamma\kappa)}) = L(G_m)$. Hence (4) follows. Furthermore, we directly see that $\hat{T}_k \subset \hat{T}_j$. The latter observation is used in demonstrating (5).

As above, if $k_j \prec k_{j+1}$, then there are $n, m \in N$, $\alpha, \beta, \gamma, \delta, \tau, \kappa \in N^*$ such that $k_j = (n, \alpha, \beta, \gamma)$ and $k_{j+1} = (m, \alpha\delta, \beta n\tau, \gamma\kappa)$. Moreover, since not $k_j \approx k_{j+1}$, we additionally have $range(\tau) \neq \{n\}$. Now suppose there is an infinite sequence $(k_j)_{j\in N}$ such that $k_j \prec k_{j+1}$ and $\bigcup_j \hat{T}_{k_j} = L$. Since $\hat{T}_k \subset \hat{T}_j$, we get in the limit a text $t$ of $L$ on which $M$ changes its mind infinitely often, a contradiction. Hence, (5) is proved.

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

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

(A) Search for the least $j \leq x$ for which $\hat{T}_j \subseteq t^+_x$. In case it is found, output $j$ and request next input. Otherwise, output nothing and request next input.

(B) Let $k$ be the hypothesis produced last by $M$ on input $t_{x-1}$, and let $y_k$ be the corresponding $y$ used to find $k$ , where $y_k = 0$, if $k$ is $M$'s first guess.

  (i) Test whether $t_x^+ \subseteq L(\hat{G}_k)$.
  In case it is, output $k$ and request next input. Otherwise, goto (ii).

  (ii) Search for the least $j \leq x$ satisfying $k \prec j$, and there is a $y_j$ such that $y_k < y_j$ as well as $t_{y_j}^+ \subseteq \hat{T}_j \subseteq t_x^+$. In case it is found, output $j$ and request next input. Otherwise, request next input and output nothing."

Since all of the $\hat{T}_j$ are uniformly recursively generable and finite and since $\prec$ is computable, we directly obtain that $M$ is an IIM. We proceed in showing that $M$ identifies $L$ monotonically on $t$.

*Claim* 1. If $M$ converges, say to $j$, then $L = L(\hat{G}_j)$.

First observe that $\hat{T}_j \subseteq L$, since otherwise $j$ cannot be any of $M$'s guesses. By (2), assertion (ii), we obtain that $L \not\subset L(\hat{G}_j)$. On the other hand, $L \setminus L(\hat{G}_j) \neq \emptyset$ would force $M$ to reject $j$ (cf. (B), test (ii)). Hence, $L = L(\hat{G}_j)$.

*Claim* 2. $M$ works monotonically.

This is an immediate consequence of property (4) and the definition of $M$.

*Claim* 3. $M$ converges on $t$.

In accordance with (2), assertion (i), one can show analogously as in the proof of Theorem 2 that $M$ outputs at least once a hypothesis. Moreover, as long as this guess is consistent with the data fed to $M$ in subsequent steps, this guess is repeated. In case $M$ finds an inconsistency, property (3) ensures that $M$ always outputs a new guess in some subsequent step. However, there might be competitive candidates that force $M$ to output a new guess before even touching the announced $j$. As long as this happens only finitely often, $M$ clearly converges, since a correct guess is never rejected. Now suppose that $M$ changes its mind infinitely often. In accordance with $M$'s definition then there is an infinite sequence $(k_j)_{j \in N}$ of all the guesses of $M$ such that $k_j \prec k_{j+1}$ for all $j$ and $\bigcup_j \hat{T}_{k_j} = L$. Hence property (5) is contradicted. This proves the theorem.

q.e.d

## 3.5. The Characterization of Dual Monotonic Inference

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

**Theorem 5.** *Let $\mathcal{L}$ be an indexed family of recursive languages. Then: $\mathcal{L} \in MON^d - TXT$ if and only if there is a space of hyptheses $\hat{\mathcal{G}} = (\hat{G}_j)_{j \in N}$, a computable relation $\prec$ over $N$, 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)* *For all* $L \in \mathcal{L}$ *and all* $k \in N$, $\hat{T}_k \subseteq L(\hat{G}_k)$.

*(3)* *For all* $L \in \mathcal{L}$ *, any* $k \in N$*, and all finite* $A \subseteq L$*, if* $\hat{T}_k \subseteq L$, $L(\hat{G}_k) \neq L$*, then there is a* $j$ *such that* $k \prec j$*, and* $A \subset \hat{T}_j \subseteq L(\hat{G}_j) = L$.

*(4)* *For all* $L \in \mathcal{L}$ *and all* $k$, $j \in N$*, if* $k \prec j$, $\hat{T}_j \subseteq L$*, then* $co - L(\hat{G}_k) \cap co - L \subseteq co - L(\hat{G}_j) \cap co - L$.

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

*Proof.* The necessity is mutatis mutandis proved analogously as in Theorem 4.

Sufficiency: The main problem we have to deal with is the detection and handling of overgeneralized hypotheses. We proceed in defining an IIM $M$ inferring $\mathcal{L}$ with respect to $\hat{\mathcal{G}}$. Let $L \in \mathcal{L}$, let $t$ be any text for $L$, and let $x \in N$. The wanted IIM $M$ is defined as follows:

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

(A) Search for the least $j \leq x$ for which $\hat{T}_j \subseteq t_x^+$. In case it is found, output $j$ and request next input. Otherwise, request next input.

(B) Let $k$ be the hypothesis produced last by $M$ on input $t_{x-1}$, and let $y_k$ be the corresponding $y$ used to find $k$ , where $y_k = 0$, if $k$ is $M$'s first guess.

(i) Execute the following tests:

- Check whether $t_x^+ \subseteq L(\hat{G}_k)$. In case it is not, goto (ii). Otherwise, continue as follows.
- Check for all $j \leq x$ satisfying $k \prec j$ and there is a $y_j$ with $y_k < y_j$ as well as $t_{y_j}^+ \subseteq \hat{T}_j \subseteq t_x^+$ whether or not there is a string $s$ with $|s| \leq max\{x, |s| \mid s \in t_x^+\}$ fulfilling $s \in L(\hat{G}_k) \setminus L(\hat{G}_j)$. In case at least one $j$ has been found, output the minimal one and request the next input. Otherwise, output $k$ and request next input.

(ii) Search for the least $j \leq x$ satisfying $k \prec j$, and there is a $y_j$ such that $y_k < y_j$ as well as $t_{y_j}^+ \subseteq \hat{T}_j \subseteq t_x^+$. In case it is found, output $j$ and request next input. Otherwise, request next input and output nothing."

We have to show that $M$ infers $L$ dual strong–monotonically.

*Claim* 1. If $M$ converges, say to $z$, then $L = L(\hat{G}_z)$.

First observe that $\hat{T}_z \subseteq L$, since otherwise $z$ cannot be any of $M$'s guesses. Suppose, $L \neq L(\hat{G}_z)$. Then we have to distinguish the following two cases:

*Case* 1. $L \setminus L(\hat{G}_z) \neq \emptyset$.
Consequently, there is a string $s \in L \setminus L(\hat{G}_z)$. Since $t$ is a text for $L$, there has to be an $x$ such that $s \in t_x^+$. Therefore $M$ has to reject $z$ in its consistency test in (i), a contradiction.
*Case* 2. $L(\hat{G}_z) \setminus L \neq \emptyset$.
Then we may assume $L(\hat{G}_z) \supset L$, since otherwise we are again in Case 1. Let $y_z$ be the corresponding $y$ used to find $z$. Since $\hat{T}_z \subseteq L$ and $L(\hat{G}_z) \neq L$, we may set $A := t_{y_z}^+$ and apply property (3). Hence, there has to be a $j$ such that $z \prec j$, $A \subset \hat{T}_j \subseteq L$ and $L(\hat{G}_j) = L$. Therefore, there must be an $x \geq j$ such that the IIM $M$ eventually finds a string $s$ satisfying $s \in L(\hat{G}_z) \setminus L(\hat{G}_j)$ as well as $|s| \leq max\{x, |s| \mid s \in t_x^+\}$. This event would force $M$ to change its mind from $z$ to $j$, a contradiction. This proves Claim 1.

*Claim* 2. $M$ works dual monotonically.

By construction, $M$ changes its mind, say from $k$ to $j$, only in case $k \prec j$. Since it additionally verifies $\hat{T}_z \subseteq t_x^+ \subseteq L$, in accordance with property (4) we immediately obtain $co - L(\hat{G}_k) \cap co - L \subseteq co - L(\hat{G}_j) \cap co - L$. Hence, $M$ works dual monotonically.

*Claim* 3. $M$ converges on $t$.

In accordance with (2), one easily shows that $M$ outputs a hypothesis at least once. This guess is repeated as long as in (B) the tests in (i) are fulfilled. Moreover, if $M$'s last guess is rejected, it always outputs a new hypothesis in some subsequent step. This is obvious as long as $M$ does not reject its last guess $k$ by verifying $t_x^+ \not\subseteq L(\hat{G}_k)$. If $M$ detects an inconsistency, then property (3) ensures that $M$ finds via (ii) a new hypothesis. Consequently, the only remaining case we have to deal with is that $M$ performs infinitely many mind changes. Let $(k_j)_{j\in N}$ be $M$'s sequence of hypotheses produced when successively fed with $t$. However, by construction we obtain the following: $k_j \prec k_{j+1}$ for all $j \in N$ and $\bigcup_j \hat{T}_{k_j} = L$. The latter equality is true since $t_{y_{k_j}}^+ \subseteq \hat{T}_{k_j}$ and $y_{k_j} < y_{k_{j+1}}$ for all $k_j$. Hence, property (5) is contradicted. This proves the claim as well as the theorem.

q.e.d.

The proof of the sufficiency in Theorem 5 yields a new approach to handling overgeneralizations when learning from positive data. This technique may also be applied to characterize $LIM - TXT$.

Next we characterize $MON^{\&} - TXT$. For that purpose, a new concept has to be elaborated which we now present. First, it is easy to argue that for any family in the class $MON^{\&} - TXT$ there exists a learner that not only satisfies the required constraints but also conservativeness and consistency (cf. Lange, Zeugmann and Kapur (1992)).

**Theorem 6.** *Let $\mathcal{L}$ be an indexed family of recursive languages. Then: $\mathcal{L} \in MON^{\&} - TXT$ if and only if there is a space of hypotheses $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in N}$, a computable relation $\prec$ over $N$, 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) For all $L \in \mathcal{L}$ and all $k \in N$,*

*(i)* $\hat{T}_k \subseteq L(\hat{G}_k)$.

*(ii)* *if* $\hat{T}_k \subseteq L$, *then* $L \not\subset L(\hat{G}_k)$.

*(3) For all* $k \in N$, *there is exactly one* $j$ *(sometimes abbreviated as* $j_k$*) such that* $j \prec k$.

*(4) For all* $L \in \mathcal{L}$ *and any* $k \in N$, *if* $\hat{T}_k \subseteq L$, *then there is a* $j'$, $j_k \prec j'$, *and* $L(\hat{G}_{j'}) = L$.

*(5) For all* $L \in \mathcal{L}$ *and any* $k$, $j \in N$, *if* $k \prec j$, $\hat{T}_j \subseteq L$, *then*

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

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

*Proof.* Necessity: Suppose first that an IIM $M$ learns $\mathcal{L}$ while satisfying the monotonic and the dual monotonic constraint as well as the consistency and conservativeness requirements with respect to some space $\mathcal{G} = (G_j)_{j \in N}$ of hypotheses. We indicate how we can define $\hat{\mathcal{G}} = (\hat{G}_j)_{j \in N}$, the relation $\prec$, and a recursively generable family $(\hat{T}_j)_{j \in N}$ of finite and non–empty sets. The procedure works as follows: Let $\sigma_1, \sigma_2, \ldots$ be an enumeration of all finite sequences (possibly with repetition) of strings from $\Sigma^*$. This enumeration is assumed to be in a canonical order so that all proper prefixes of a sequence appear before the sequence itself. A function $f$ that takes a sequence $\sigma$ as its argument and returns an element of $N$ is used. We also need to keep track of sequences which have already been dealt with. These are said to be *marked*. Initially, all sequences are unmarked. For any $r \geq 1$, in order to define $\hat{T}_r$, it first needs to define $\hat{T}_j$ for all $j < r$. Suppose the procedure has already defined $\hat{T}_j$ for all $j < r$. In order to define $\hat{T}_r$, a dovetailing technique is invoked. By $c : N \times N \to N$, we denote Cantor's pairing function. Also, *first* and *second* are the two inverse functions, so that $\mathit{first}(c(x, y)) = x$ and $\mathit{second}(c(x, y)) = y$. The procedure looks for the least number $k$ such that $\sigma_{\mathit{first}(k)}$ is an unmarked sequence contained in the language $L_{\mathit{second}(k)} \in \mathcal{L}$. It then marks this sequence and runs $M$ on it. If $M$ produces a hypothesis $u$ when fed the entire sequence $\sigma_{\mathit{first}(k)}$, then it lets $\hat{T}_r = \sigma^+_{\mathit{first}(k)}$ and $\hat{G}_r = G_u$; otherwise, it again looks for the first unmarked sequence as above. If $M$ has produced this guess for the first time on seeing the entire sequence $\sigma_{\mathit{first}(k)}$, it lets $f(\sigma_{\mathit{first}(k)}) = r$. If the sequence $\sigma_i$ is the least prefix of $\sigma_{\mathit{first}(k)}$ on which the machine produced a guess, it lets $f(\sigma_i) \prec r$.

It is easy to see that the procedure outlined above is well-defined. Since the sequences are assumed to be enumerated in a canonical order and proper dove-tailing is used, all the prefixes of a sequence would be marked before the sequence itself is marked. Thus, the function $f$ would always be defined on a certain value prior to its use on that value. Since the family is learned by $M$, there would be an infinite number of different sequences which are contained in some language in the family and on which $M$ makes a guess. Thus, the procedure constructively defines $\hat{T}_r$ and $\hat{G}_r$ for each $r \geq 1$. We next argue that the properties (1) through (5) are also satisfied.

Property (1) is trivial to establish since exactly whenever the machine $M$ makes any output $u$, there is some $\hat{G}_r$ defined equal to $G_u$. The second property is met since a conservative learner must always guess a least upper bound of the evidence. Furthermore, property (3) is also satisfied because whenever a sequence $\sigma_i$ leads to a definition of a $\hat{T}_r$, by construction, there is a unique $j_r$ such that $j_r \prec r$.

Suppose there is some $L \in \mathcal{L}$ and a $k \in N$, such that $\hat{T}_k \subseteq L$. Then, by construction, there must be a sequence $\sigma_i$ such that $\sigma_i^+ = \hat{T}_k$ and $\hat{G}_k = G_{M(\sigma_i)}$. Let $t$ be a text for $L$ that has $\sigma_i$ as a prefix. Clearly, at some point, the machine when run on $t$ must guess some $j'$ so that $L = L(\hat{G}_{j'})$. Since $j_k = f(\sigma_k)$, where $\sigma_k$ is the least prefix of $\sigma_i$ on which $M$ makes a guess, we immediately have $j_k \prec j'$. Thus, the fourth property is also satisfied.

Finally, we show that if, for any $k, j \in N$ and $L \in \mathcal{L}$, $k \prec j$, $\hat{T}_j \subseteq L$, then

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

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

By construction, $k \prec j$ if and only if there is a sequence $\sigma_i$ ($\sigma_i^+ = \hat{T}_j$) such that when $M$ is run on $\sigma_i$, the machine outputs a guess $u$ as its first guess such that $G_u = \hat{G}_k$. Further, on seeing the entire sequence $\sigma_i$, it outputs some $v$ such that $G_v = \hat{G}_j$. Let any language $L \in \mathcal{L}$ be such that $\sigma_i^+ \subseteq L$. Then, the sequence $\sigma_i$ can be extended by any text $t$ for $L$ so that the entire presentation is a text for $L$. Clearly, the machine $M$ must converge to an index for $L$ on such a text. We claim that $L(\hat{G}_j) \setminus L(\hat{G}_k) \subseteq L$. Suppose the converse, i.e., there is a string $s$ such that $s \in L(\hat{G}_j) \setminus L(\hat{G}_k)$ and $s \notin L$. Hence, $s \in co - L \cap co - L(\hat{G}_k)$ but $s \notin co - L \cap co - L(\hat{G}_j)$. Therefore, $co - L(\hat{G}_k) \cap co - L \not\subseteq co - L(\hat{G}_j) \cap co - L$ and consequently, $M$ does not work dual monotonically. Thus, (i) is proved. We proceed in showing (ii). Suppose, 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 $s \notin L(\hat{G}_j)$. Therefore, $s \notin L(\hat{G}_j) \cap L$. However, the latter statement contradicts the monotonicity requirement $L(\hat{G}_k) \cap L \subseteq L(\hat{G}_j) \cap L$. Hence, (ii) is proved.

Sufficiency: Suppose we are given a space of hypotheses $\hat{\mathcal{G}} = (\hat{G}_j)_{j \in N}$ , a computable relation $\prec$ and a recursively generable family $(\hat{T}_j)_{j \in N}$ . We construct an IIM $M$ that learns $\mathcal{L}$ as follows: Let $L \in \mathcal{L}$, let $t$ be any text for $L$ and let $x \in N$. $M$ on input $t_x$ behaves differently depending on whether or not it has made a guess prior to this stage.

- If $M$ has not produced a guess up to this stage, then scan the enumeration of $(\hat{T}_j)_{j \in N}$ for the least $k \leq x$ such that $\hat{T}_k \subseteq t_x^+ \subseteq L(\hat{G}_k)$. If such a $k$ is found, then find $j_k$ (the only $j$ such that $j \prec k$) and let $\mathit{FIRST} = j_k$, output $k$, and request the next input. Otherwise, do not output a guess, and request the next input.

- If $M$ has produced a guess previous to this stage, maintain the previous guess if it is consistent with $t_x^+$. Otherwise, scan the enumeration of $(\hat{T}_j)_{j \in N}$ for the least $k \leq n$ such that $j_k = \mathit{FIRST}$ and $\hat{T}_k \subseteq t_x^+ \subseteq L(\hat{G}_k)$. If such a $k$ is found, then output it as the guess, and request the next input. Otherwise, output nothing, and request the next input.

We first argue that $M$ learns $\mathcal{L}$. Let $L \in \mathcal{L}$ and let $t = t_1, t_2, \ldots$ be a text for $L$. The machine $M$, when run on $t$, must make a non-zero guess since, by property (1), there must be a least $k$ such that $L = L(\hat{G}_k)$ and, at some stage $x \geq k$, the text must include the finite set $\hat{T}_k$. Since the machine is conservative and consistent (by property (ii) of (2) and by construction) it never produces an overgeneralization of the language to be learnt. Moreover, a correct guess is never rejected. Hence, the only way the machine could fail to converge to an index for $L$ is by guessing an infinite number of guesses all different from $L$. By property (4), there must be a least $j^*$ such that $\mathit{FIRST} \prec j^*$ and $L = L(\hat{G}_{j^*})$. Consider a stage at which all of $\hat{T}_{j^*}$ has been witnessed in the evidence. Suppose, at some subsequent stage, the machine changes its guess to $j$. We claim that $L(\hat{G}_j) = L(\hat{G}_{j^*})$. Clearly, $(\hat{T}_j \cup \hat{T}_{j^*}) \subseteq (L(\hat{G}_j) \cap L(\hat{G}_{j^*}))$. Since $\mathit{FIRST} \prec j^*$, by property (5), we have $L(\hat{G}_{j^*}) \setminus L(\hat{G}_{\mathit{FIRST}}) \subseteq L(\hat{G}_j)$. Likewise, since $\mathit{FIRST} \prec j$, we have $L(\hat{G}_j) \setminus L(\hat{G}_{\mathit{FIRST}}) \subseteq L(\hat{G}_{j^*})$. For the same reason, we have $(L(\hat{G}_{\mathit{FIRST}}) \setminus L(\hat{G}_{j^*})) \cap L(\hat{G}_j) = \emptyset$ and $(L(\hat{G}_{\mathit{FIRST}}) \setminus L(\hat{G}_j)) \cap L(\hat{G}_{j^*})) = \emptyset$. Thus, $L(\hat{G}_j) = L(\hat{G}_{j^*})$. Since $M$ is conservative, we conclude that $M$ converges to an index for $L$ on $t$.

Suppose on a text $t$ for a language $L$, the machine makes the guesses $i, \ldots, u, \ldots, v, \ldots, j$, and it is the case that the machine exhibits a violation of monotonicity or dual monotonicity while proceeding from the guess $u$ to $v$. Therefore,

(c.1) $L(\hat{G}_v) \setminus L(\hat{G}_u) \not\subseteq L$ (violation of dual monotonicity), or

(c.2) $(L(\hat{G}_u \setminus L(\hat{G}_v)) \cap L \neq \emptyset$ (violation of monotonicity).

¿From the construction of $M$, we can infer that, for $\mathit{FIRST} = j_i$, both $\mathit{FIRST} \prec u$ and $\mathit{FIRST} \prec v$, where $\hat{T}_u \subseteq L(\hat{G}_v)$ and $\hat{T}_u \cup \hat{T}_v \subseteq L$. Since $\mathit{FIRST} \prec u$ as well as $\hat{T}_u \subseteq L(\hat{G}_v)$, and $L(\hat{G}_v) \in \mathcal{L}$, due to property (5), we have

(d.1) $L(\hat{G}_u) \setminus L(\hat{G}_{\mathit{FIRST}}) \subseteq L(\hat{G}_v)$ and

(d.2) $(L(\hat{G}_{\mathit{FIRST}}) \setminus L(\hat{G}_u)) \cap L(\hat{G}_v) = \emptyset$.

Similarly, since $\mathit{FIRST} \prec v$, due to property (5), we have

(e.1) $L(\hat{G}_v) \setminus L(\hat{G}_{\mathit{FIRST}}) \subseteq L$ and

(e.2) $(L(\hat{G}_{\mathit{FIRST}}) \setminus L(\hat{G}_v)) \cap L = \emptyset$.

Suppose (c.1) is true, i.e., there is some string $x \in L(\hat{G}_v)$ which is neither in $L(\hat{G}_u)$ nor $L$. By (e.1), we can conclude that $x \in L(\hat{G}_{\mathit{FIRST}})$. But this is inconsistent with (d.2). Suppose, on the other hand, that (c.2) is true. Therefore, there is some string $x$ in $L(\hat{G}_u)$ which is also in $L$ but not in $L(\hat{G}_v)$. By (e.2), we can conclude that $x \notin L(\hat{G}_{\mathit{FIRST}})$. But this is inconsistent with (d.1). This contradicts our assumption that $M$ violates monotonicity or dual monotonicity.

q.e.d.

Note that the above characterizations could be generalized to all types of monotonic learning and finite inference on informant. This is done in Lange and Zeugmann (1992B).

# 4. Conclusions and Open Problems

Different notions of monotonicity and of dual monotonicity have been defined and almost all resulting modes of inference from positive data have been characterized in terms of finitely generable tell–tale sets. In particular, the characterization of $WMON - TXT$ solved the long standing open problem of how to characterize inference algorithms that avoid overgeneralization. Moreover, in characterizing dual monotonic inference ¿from text, we developed a new method to detect and to deal with overgeneralizations. This technique can also be applied to obtain a new characterization of language learning in the limit from positive data. However, the characterization of dual weak–monotonic inference remains open.

All these characterization theorems lead to a deeper insight into the problem what actually may be inferred monotonically. Moreover, we obtained a unifying approach to monotonic language learning in describing general algorithms that perform any monotonic inference task. Furthermore, the characterization theorems may eventually be applied to solve problems that could not be solved using other approaches. In order to have an example, let us recall what we have derived from Theorem 2, i.e., 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. Nevertheless, while the decidability of set inclusion of languages is necessary for $SMON - TXT$ identification, in general it is not sufficient. In Lange and Zeugmann (1991) we have shown that there is an indexed family of recursive languages such that set inclusion is uniformly decidable but which is not *monotonically* inferable, even on *informant*.

However, several problems remained open. One of the most intriguing questions is whether or not all types of monotonic and of dual monotonic inference from positive data may be performed by IIMs that are rearrangement–independent, or even set–driven. For strong–monotonic inference, this question has been partially answered via the characterization theorem. Unfortunately, for weak–monotonic, monotonic and dual monotonic language learning, this approach did not succeed. Nevertheless, we were able to characterize rearrangement–independent monotonic inference from positive data (denoted by $MONR - TXT$) as follows:

**Theorem** *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 $L(\hat{G}_k) \cap L \subseteq L(\hat{G}_j) \cap L$.*

Obviously, we have $MONR - TXT \subseteq MON - TXT$. Therefore, clarifying whether the inclusion is proper either yields a simplified characterization of $MON - TXT$ or it adds some evidence that Theorem 4 cannot be considerably improved. Note that it is not hard to show that $SMON - TXT \subset MONR - TXT$ (cf. Lange and Zeugmann (1991)).

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 of pairs (argument,value) 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 language learning have special features distinguishing them from monotonic inference of recursive functions. Therefore, it would be very interesting to study monotonic language learning in the general case, i.e., not restricted to indexed families.

**Acknowledgement**

The first two authors gratefully acknowledge many enlightening discussions with Rolf Wiehagen concerning the characterization of learning algorithms.

# 5. References

Angluin, D. (1980), Inductive Inference of Formal Languages from Positive Data, *Information and Control* **45**, 117 - 135.

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

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

BARZDIN, YA.M. (1974), Inductive Inference of Automata, Functions and Programs, *in* "Proceedings International Congress of Math.," Vancouver, pp. 455 - 460.

BLUM, L., AND BLUM, M. (1975), Toward a Mathematical Theory of Inductive Inference, *Information and Control* **28**, 122 - 155.

BUCHER, W., AND MAURER, H. (1984), "Theoretische Grundlagen der Programmiersprachen, Automaten und Sprachen," Bibliographisches Institut AG, Wissenschaftsverlag, Zürich.

CASE, J. (1988), The Power of Vacillation, *in* "Proceedings 1st Workshop on Computational Learning Theory," (D. Haussler and L. Pitt, Eds.), pp. 196 -205, Morgan Kaufmann Publishers Inc.

CASE, J., AND LYNES, C. (1982), Machine Inductive Inference and Language Identification, *in* "Proceedings Automata, Languages and Programming, Ninth Colloquim, Aarhus, Denmark," (M. Nielsen and E.M. Schmidt, Eds.), Lecture Notes in Computer Science Vol. 140, pp. 107 - 115, Springer-Verlag, Berlin.

FREIVALDS, R., KINBER, E.B., AND WIEHAGEN, R. (1989), Inductive Inference from Good Examples, *in* "Proceedings International Workshop on Analogical and Inductive Inference, October 1989, Reinhardsbrunn Castle," (K.P. Jantke ,Ed.), Lecture Notes in Artificial Intelligence Vol. 397, pp.1 - 17, Springer-Verlag, Berlin.

FREIVALDS, R., KINBER, E.B., AND WIEHAGEN, R. (1992), Convergently versus Divergently Incorrect Hypotheses in Inductive Inference, GOSLER Report 02/92, January 1992, Fachbereich Mathematik und Informatik, TH Leipzig.

FULK, M.(1990), Prudence and other Restrictions in Formal Language Learning, *Information and Computation* **85**, 1 - 11.

GOLD, M.E. (1967), Language Identification in the Limit, *Information and Control* **10**, 447 - 474.

JAIN, S., AND SHARMA, A. (1989), Recursion Theoretic Characterizations of Language Learning, The University of Rochester, Dept. of Computer Science, TR 281.

JANTKE, K.P. (1991A), Monotonic and Non–monotonic Inductive Inference, *New Generation Computing* **8**, 349 - 360.

JANTKE, K.P. (1991B), Monotonic and Non–monotonic Inductive Inference of Functions and Patterns, *in* "Proceedings 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 Vol. 543, pp. 161 - 177, Springer-Verlag, Berlin.

Kapur, S. (1992B), Monotonic Language Learning, *in* "Proceedings of the Workshop on Algorithmic Learning Theory," JSAI.

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

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

Lange, S., and Zeugmann, T. (1991), Monotonic versus Non-monotonic Language Learning, *in* "Proceedings 2nd International Workshop on Nonmonotonic and Inductive Logic, December 1991, Reinhardsbrunn," to appear in Lecture Notes in Artificial Intelligence, Springer-Verlag, Berlin.

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

Lange, S., and Zeugmann, T. (1992B), Characterization of Language Learning on Informant under Various Monotonicity Constraints, *Journal of Experimental and Theoretical Artificial Intelligence*, to appaer.

Lange, S., and Zeugmann, T. (1992C), Learning Recursive Languages with Bounded Mind Changes, GOSLER–Report 16/92, FB Mathematik und Informatik, TH Leipzig.

Lange, S.,Zeugmann, T., and Kapur, S (1992), Class Preserving Monotonic Language Learning, submitted to *Theoretical Computer Science*, and GOSLER–Report 14/92, FB Mathematik und Informatik, TH Leipzig.

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.

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

Porat, S., and Feldman, J.A. (1988), Learning Automata from Ordered Examples, *in* "Proceedings First Workshop on Computational Learning Theory," (D. Haussler and L. Pitt, Eds.), pp. 386 - 396, Morgan Kaufmann Publishers Inc.

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

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

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

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

Wiehagen, R. (1991), A Thesis in Inductive Inference, *in* "Proceedings 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 Vol. 543, pp. 184 - 207, Springer-Verlag, Berlin.

Wiehagen, R. (1992), Personal Communication

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

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