Types of Monotonic Language Learning and Their Characterization

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

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

Abstract

The present paper deals with strong–monotonic, monotonic and weak–monotonic language learning from positive data as well as from positive and negative examples. The three notions of monotonicity reflect different formalizations of the requirement that the learner has to produce always better and better generalizations when fed more and more data on the concept to be learnt. We characterize strong–monotonic, monotonic, weak–monotonic and finite language learning from positive data in terms of recursively generable finite sets, thereby solving a problem of Angluin (1980). Moreover, we study monotonic inference with iteratively working learning devices which are of special interest in applications. In particular, it is proved that strong–monotonic inference can be performed with iteratively learning devices without limiting the inference capabilities, while monotonic and weak–monotonic inference cannot.

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 today state of the art is excellently surveyed in Angluin and Smith (1983, 1987).

[*]Most of this work has been performed while the author was visiting TH Darmstadt.

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, as well as both *positive and negative examples*, i.e., the learner is fed with arbitrary strings over the underlying alphabet which are classified with respect to their containment to the unknown language. The sequence of hypotheses has to converge to a hypothesis correctly describing the object to be learnt. Monotonicity requirements have been introduced by Jantke (1991A, 1991B) and Wiehagen (1991). In Lange and Zeugmann (1991, 1992) we have dealt with monotonic language learning of indexed families. The main underlying question can be posed as follows: Would it be possible to infer the unknown language in a way such that *only* better and better hypotheses are output? The strongest interpretation of this requirement means that we are forced to produce an augmenting chain of languages, i.e., $L_i \subseteq L_j$ iff L_j is guessed later than L_i (cf. Definition 3 (A)).

Wiehagen (1991) proposed to interpret "better" with respect to the language L having to be learnt, i.e., now we require $L_i \cap L \subseteq L_j \cap L$ iff L_j appears later in the sequence of guesses than L_i does (cf. Definition 3 (B)). That means, a new hypothesis is never allowed to destroy something what a previously generated guess already *correctly* reflects.

The third version of monotonicity, which we call weak–monotonicity, is derived from non–monotonic logics and adopts the concept of cumulativity. Hence, we only require $L_i \subseteq L_j$ as long as there are no data fed to the inference device after having produced L_i that contradict L_i (cf. Definition 3 (C)).

In all what follows we restrict ourselves to deal exclusively with the learnability of indexed families of non–empty uniformly recursive languages. This case is of special interest with respect to potential applications. The first problem arising naturally is to relate all types of monotonic language learning one to the other as well as to previously studied modes of inference. This question has been completely answered in Lange and Zeugmann (1991, 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 if the former hypothesis "provably misclassifies" some word with respect to the data seen so far. Considering learning from positive and negative examples in the setting of indexed families it is not hard to prove that conservativeness does not restrict the inference capabilities. Surprisingly enough, in the 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)). 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 in characterizing all types of monotonic language learning as well as of finite inference of recursive languages in terms of recursively generable families of recursive, non–empty and finite sets. Very recently, Kapur and Bilardi(1992) also established a characterization of conservative learning. However, their characterization differs at least conceptually from ours. In order to see what the difference is 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 total learning algorithms such that in each pair the algorithm accepts a language of the family $\mathcal{L}$ being a least upper bound of the corresponding finite set, and for each $L \in \mathcal{L}$ there is at least a corresponding pair. Consequently, their characterization is conceptually based on the judicious use of a function computing least upper bounds. Our approach is in some sense conversely in that we construct a suitable enumeration $\hat{\mathcal{L}}$ of $\mathcal{L}$ and for 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.

The second part of the paper deals with monotonic inference that is performed by *iteratively* working machines (cf. Wiehagen (1976)). With respect to potential applications iterative learning is highly desirable, since in order to compute the next guess it uses *only* the last hypothesis and the next string of the language to be learnt. Hence that model of learning takes into account the limitation of space in all realistic computations. Recently, several iterative language learning algorithms have been published (cf. e.g. Porat and Feldman (1988), Lange and Wiehagen (1991)). As it turned out, in general machines working both, iteratively and monotonically are less powerful than monotonic ones. However, strong–monotonic inference from positive data can always be achieved by iteratively working machines.

2 Preliminaries

By $N = \{1, 2, 3, ...\}$ we denote the set of all natural numbers. In the sequel we assume familarity with formal language theory (cf. e.g. Bucher and Maurer (1984)). By Σ we denote any fixed finite alphabet of symbols. Let Σ^* be the free monoid over Σ. The length of a string $w \in \Sigma^*$ is denoted by $|w|$. Any subset $L \subseteq \Sigma^*$ is called a language. By $co - L$ we denote the complement of L, i.e., $co - L = \Sigma^* \setminus L$. Let L be a language and $t = s_1, s_2, s_3, ...$ a sequence of strings from Σ^* such that $range(t) = \{s_k \mid k \in N\} = L$. Then t is said to be a *text* for L or, synonymously, a *positive presentation*. Furthermore, let $i = (s_1, b_1), (s_2, b_2), ...$ be a sequence of elements of $\Sigma^* \times \{+, -\}$ such that $range(i) = \{s_k \mid k \in N\} = \Sigma^*$, $i^+ = \{s_k \mid (s_k, b_k) = (s_k, +), k \in N\} = L$ and $i^- = \{s_k \mid (s_k, b_k) = (s_k, -), k \in N\} = co - L$. Then we refer to i as an *informant*. If L is classified via an informant then we also say that L is represented by positive and negative data. Moreover, let t, i be a text and an informant, respectively, and let x be a number. Then t_x, i_x denote the initial segment of t and i of length x, respectively, e.g., $i_3 = (s_1, b_1), (s_2, b_2), (s_3, b_3)$. Let t be a text and let $x \in N$. 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 $w \in \Sigma^*$ we have

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

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

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

At this point we have to clarify what space of hypotheses we should choose, thereby also specifying the goal of the learning process. Gold (1967) and Wiehagen (1977) pointed out that there is a difference in what can be inferred in dependence on whether we want to synthesize in the limit grammars (i.e., procedures generating languages) or decision procedures, i.e., programs of characteristic functions. Case and Lynes (1982) investigated this phenomenon in detail. As it turns out, IIMs synthesizing grammars can be more powerful than those ones which are requested to output decision procedures. However, in the context of identification of indexed families both concepts are of equal power. Nevertheless, we decided to require the IIMs to output grammars. This decision has been caused by the fact that there is a big difference between the possible monotonicity requirements. A straightforward adaptation of the approaches made in inductive inference of recursive functions directly yields analogous requirements with respect to the corresponding characteristic functions of the languages to be inferred. On the other hand, it is only natural to interpret monotonicity with respect to the language to be learnt, i.e., to require containement of languages as described in the introduction. As it turned out, the latter approach increases considerably the power of monotonic language learning. Furthermore, since we exclusively deal with indexed families $\mathcal{L} = (L_j)_{j\in N}$ of recursive languages we almost always take as space of hypotheses an enumerable family of grammars $G_1, G_2, G_3, ...$ over the terminal alphabet Σ satisfying $\mathcal{L} = \{L(G_j) \mid j \in N\}$. Moreover, we require that membership in $L(G_j)$ is uniformly decidable for all $j \in N$ and all strings $w \in \Sigma^*$. As it turns out, it is sometimes very important to choose the space of hypotheses appropriately in order to achieve the desired learning goal. Then the IIM outputs numbers j which we interpret as G_j.

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

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

Definition 1 could be easily generalized to arbitrary families of recursively enumerable languages (cf. Osherson et al. (1986)). Nevertheless, we exclusively consider the restricted case defined above, since our motivating examples are all indexed families of recursive languages. Note that, in general, it is not decidable whether or not M has already inferred L. In case M produces only a *single* and *correct* guess after having been fed an initial segment of a text t (or informant i) and *stops* then, we say that M *finitely* infers L on t (on i). M $FIN-TXT$ $(FIN-INF)$–infers L iff it finitely infers L on every text (informant).
The resulting identification type is denoted by $FIN-TXT$ $(FIN-INF)$.

Next we want to formally define strong–monotonic, monotonic and weak–monotonic inference. But before doing this, first we 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 been already fed with.

Definition 2, Barzdin ((1974)) *An IIM M $CONS-TXT$ $(CONS-INF)$–identifies L on a text t (an informant i) iff*

(1) M $LIM-TXT$ $(LIM-INF)$–identifies L on t (on i)
(2) Whenever M on t_x (i_x) produces a hypothesis j_x then $range(t_x) \subseteq L(G_{j_x})$ $(i_x^+ \subseteq L(G_{j_x})$ 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 $CONS-TXT(M)$ $(CONS-INF(M))$ we denote the set of all languages which M does $CONS-TXT$ $(CONS-INF)$–identify. $CONS-TXT$ and $CONS-INF$ are analogously defined as above.

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

(A) strong–monotonically
(B) monotonically
(C) weak–monotonically
iff

M $LIM-TXT$ $(LIM-INF)$–identifies L and for any text t (informant i) of L as well as for any two consecutive hypotheses j_x, j_{x+k} which M has produced when fed t_x and t_{x+k} (i_x and i_{x+k}), for some $k \geq 1, k \in N$, the following conditions are satisfied:

(A) $L(G_{j_x}) \subseteq L(G_{j_{x+k}})$
(B) $L(G_{j_x}) \cap L \subseteq L(G_{j_{x+k}}) \cap L$
(C) if $t_{x+k} \subseteq L(G_{j_x})$ then $L(G_{j_x}) \subseteq L(G_{j_{x+k}})$ (if $i_{x+k}^+ \subseteq L(G_{j_x})$ and $i_{x+k}^- \subseteq co-L(G_{j_x})$, then $L(G_{j_x}) \subseteq L(G_{j_{x+k}})$).

We denote by $SMON-TXT$, $SMON-INF$, $MON-TXT$, $MON-INF$, $WMON-TXT$, $WMON-INF$ the family of all thoses sets $\mathcal{L}$ of indexed families of languages for which there is an IIM inferring it strong–monotonically, monotonically, and weak–monotonically from text t or informant i, respectively.
Note that even $SMON-TXT$ contains interesting "natural" families of formal languagues (cf. Lange and Zeugmann (1991, 1992)). Finally in this section we define *conservatively* working IIMs.

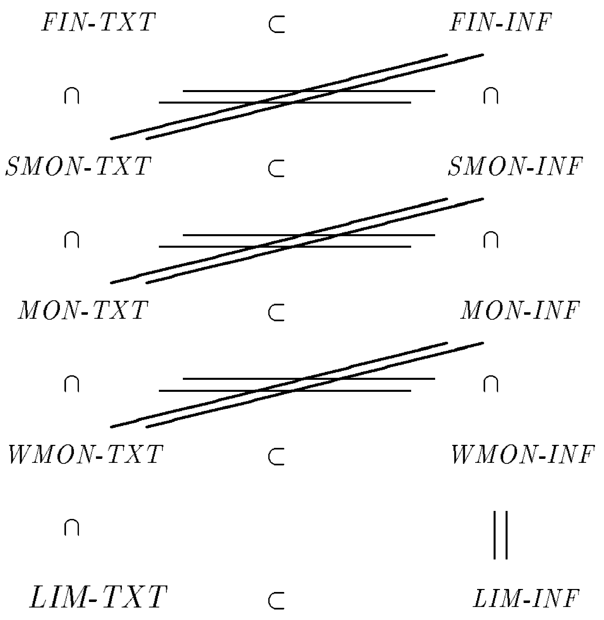
Definition 4, (Angluin (1980A))
An IIM M CONSERVATIVE–TXT (CONSERVATIVE–INF)–identifies L on text t (on informant i), iff for every text t (informant i) the following conditions are satisfied:

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

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

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

Intuitively speaking, a conservatively working IIM performs *exclusively* justified mind changes. Note that $WMON-TXT =$ *CONSERVATIVE–TXT* as well as $WMON-INF =$ *CONSERVATIVE–INF.*



(* # denotes incomparability of sets. *) Finally, the figure above summarizes the known results concerning monotonic inference (cf. Lange and Zeugmann (1991, 1992)).

3 Characterization Theorems

In this section we give characterizations of strong–monotonic, monotonic and weak–monotonic inference from positive data as well as for $FIN-TXT$. Characterizations play an important role in inductive inference in that they lead to a deeper insight into the problem how algorithms performing the inference process may work (cf. e.g. Wiehagen (1977, 1991), Angluin (1980), Freivalds et al. (1992)). 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 $c : N \times N \rightarrow N$ 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 $s_1, s_2, \ldots$ be the lexicographically ordered text of Σ^*. Test sequentially whether $s_z \in L(G_k)$ for $z = 1, 2, 3, \ldots$ 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 $t^k_1 = s_z$. We proceed inductively:

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

We define:

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

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 our 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 $L(\hat{G}_z) \subset L(\hat{G}_j)$. 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 too, 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. By construction the machine M works conservatively, since it changes its mind only in case it finds an inconsistency in (A).

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 an hypothesis. Furthermore, after having fed t_x to M, the machine M always outputs an 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.

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

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

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

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

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

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 anologously 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 σ 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 furthermore 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 to 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 remain 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 works indeed strong–monotonically on t.

q.e.d.

The latter theorems have some interesting consequences which shall discuss below. Now we present the announced characterization of $FIN-TXT$ and postpone that one for $MON-TXT$ for a moment, since it deserves special attention. Note that an analogous theorem has been obtained independently by Mukouchi (1991).

However, even the next theorem has some special features distinguishing it from the characterizations already given. As pointed out above, dealing with characterizations has been motivated by the aim to elaborate a unifying approach to monotonic inference. Concerning $WMON-TXT$ 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. As the results of Angluin (1980) show, this choice is inevitable. The next theorem yields even a stronger implication. Namely, it shows, if there is a space of hypotheses at all such that $\mathcal{L} \in FIN-TXT$ 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 $L_j = L_k$.

Finally in this section 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 textsegment fed to M one has to deal with a *non–recursive* component. The non–recursiveness directly results from the requirement that M has to infer each $L \in \mathcal{L}$ from any text, i.e., one has to find suitable approximations of the uncountable many non–recursive texts. 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 hypotheses 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 to property (3) yields the following surprising 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 ones.

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 totally differs 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 are 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 $k \in N$,

(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 $k \prec j$, $A \subseteq \hat{T}_j \subseteq L(\hat{G}_j) = L$.

(4) For all $L \in \mathcal{L}$ and all $k,\ j \in N$,

(i) if $k \prec j$, then $\hat{T}_k \subset \hat{T}_j$.

(ii) 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 α. Whenever appropriate we interpret a number k as bijective encoding of a 4-tupel (n,α,β,γ), 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 $k = (n,\alpha,\beta,\gamma)$, $j = (m,\alpha\delta,\beta n\tau$, 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 (1992)). 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 $(|\alpha| = |\beta| = |\gamma|)$, $\beta = \hat{\beta}n\rho$, $n \notin range(\hat{\beta})$, then $\tilde{T}_{(n,\alpha,\beta,\gamma)} = \emptyset$ if $range(\rho) \neq \{n\}$.

(iii) Otherwise let $\alpha = y_1,...y_r$, $\beta = 0n_1,...n_{r-1}$, $\gamma = z_1,...z_r$ 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_1th element σ_{n_1} (with respect to the lexicographical ordering) of $visible(\hat{\sigma}_{n_1})$ when fed successively to M exactly yields the sequence $0n_1$ of hypotheses. If not, then $\tilde{T}_{(n,\alpha,\beta,\gamma)} = \emptyset$, and stop.

In case it does, generate the canonical text $\hat{\sigma}_{n_2}$ of $L(G_{n_2})$ of lenght $y_1 + y_2$. 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_2th element σ_{n_2} (with respect to the lexicographical ordering) of $visible(\hat{\sigma}_{n_2})$ does fulfil the following properties:

(a) σ_{n_1} is a prefix of σ_{n_2} , and

(b) When fed successively with σ_{n_2} the machine M exactly produces the sequence $0n_1n_2$ 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 $y_1 + ... + y_n$ and compute $visible(\hat{\sigma}_n)$. If $|visible(\hat{\sigma}_n)| < z_n$, then set $\tilde{T}_{(n,\alpha,\beta,\gamma)} = \emptyset$. Else let σ_n be the z_nth element of $visible(\hat{\sigma}_n)$ with respect to the lexicographical ordering. Test whether

(a) σ_{n_r} is a prefix of σ_n

(b) When fed successively with σ_n the machine M exactly produces the sequence $0n_1n_2...n_rn$ 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 β to be the sequence $On_1n_2...n_r$ 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 textsegments 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 lenght $a_i = |\sigma_{n_1}| + i$, $i = 0, 1, ...$ 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 σ_{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 lenght $a_i = |\sigma_{n_2}| + i$ 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 σ_{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. 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 $visible(t^L_{y_{r+1}})$. The $\alpha = y_1...y_{r+1}$, $\beta = 0n_1...n_r$ and $\gamma = z_1...z_{r+1}$ directly yield $\tilde{T}_{(n,\alpha,\beta,\gamma)} \neq \emptyset$. 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 $\alpha = y_1...y_r$, $\beta = 0n_1...n_{r-1}$, $\gamma = z_1...z_r$, $q = y_1 + ... + y_r$, 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 β 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.

The part (i) of property (4) is an immediate consequence of our construction. For showing (ii) recall the definition of the relation $\prec$. Since $k \prec j$ 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 textsegment σ 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, part (ii) of property (4) follows.

We continue in proving (5). Again, recall the definition of the relation $\prec$. 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 $\bigcup_j \hat{T}_{k_j} = L$. That means, in the limit we get 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, 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 competive candidates forcing 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

Note that the above characterizations could be generalized to all types of monotonic learning and finite inference on informant. This is done in a subsequent paper. In the next section we consider a problem which is important with respect to potential applications of learning algorithms. We ask whether the inference process can be performed with *IIM*s that only use the last guess and the next string of the text resp. informant of the language to be learnt as their input.

4 Monotonic Inference with Iteratively Working IIM

Conceptionally, an iteratively working *IIM* M defines a sequence $(M_n)_{n \in N}$ of machines each of which takes as its input the output of its predecessor. Hence, the *IIM* M has always to produce a hypothesis, i.e., it is always in the "or–case".

Definition 5, Wiehagen (1976) *An IIM M $IT - TXT$ ($IT - INF$)-identifies L on a text $t = (s_j)_{j \in N}$ (an informant $i = ((w_j, b_j))_{j \in N}$) iff*

(1) For all $n \in N$, $M_n(t)$ ($M_n(i)$) is defined, where $M_1(t) := M(s_1)$ ($M_1(i) := M((w_1, b_1))$) and $M_{n+1}(t) := M(M_n(t), s_{n+1})$ ($M_{n+1}(i) := M(M_n(i), (w_{n+1}, b_{n+1}))$).

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

M $IT - TXT$ ($IT - INF$)-identifies L iff M $IT - TXT$ ($IT - INF$)-identifies L on every text t (informant i). The resulting identification types $IT - TXT$ and $IT - INF$ are analogously defined as above.

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

Since iteratively *IIMs* are always required to produce an output, a relaxation with respect to the space of hypotheses $\mathcal{G} = (G_j)_{j \in N}$ is appropriate, i.e., we weaken the condition $\mathcal{L} = \{L(G_j) \mid j \in N\}$ to $\mathcal{L} \subseteq \{L(G_j) \mid j \in N\}$. However, we further require that membership in $L(G_j)$ is uniformly decidable. The next theorem relates iterative inference on informant to monotonic ones.

Theorem 5.

(1) $IT - INF \subset WMON - INF$

(2) $IT - INF \# MON - INF$

(3) $WMON - IT - INF \subset WMON - INF$

(4) $MON - IT - INF \subset MON - INF$

Proof. The proof of the theorem is done via the following lemmata.

Lemma A. $MON-INF \setminus IT-INF \neq \emptyset$

Proof (Lemma A). Let $(I_j)_{j\in N}$ be a canonical bijective and computable enumeration of all finite sets of natural numbers. We define the wanted indexed family $\mathcal{L}$ over the alphabet $\{a,b\}$ as follows: $L_1 = \{a\}^*$, $L_2 = \{b\}^*$, and for all $j \in N$ we set $L_{j+2} = L(I_j) = L_1 \cup (L_2 \setminus \{b^n \mid n \in I_j\})$. Then we define $\mathcal{L} = \{L_{j+2} \mid j \in N\} \cup \{L_1, L_2\}$. Obviously, $\mathcal{L}$ is an indexed family of non–empty uniformly recursive languages.

Claim 1: $\mathcal{L} \in MON-INF$

Let $L \in \mathcal{L}$, and let i be any informant for L. For technical convenience, let $L_0 = \emptyset$. The wanted IIM M inferring $\mathcal{L}$ with respect to $\mathcal{L}$ is informally defined as follows:

Initially it outputs 0. This guess is repeated until M receives the first string $(w,+)$. Now we distinguish two cases:

Case 1: $w \in \{a\}^*$

Then M outputs 1. As long as M obtains $(w,+) \in \{a\}^* \times \{+,-\}$, it repeats its guess. If in i_x appears the first $(w,+) \in \{b\}^* \times \{+,-\}$, then M computes $I = \{k \mid b^k \in i_x^-\}$ as well as I's index in the enumeration $(I_j)_{j\in N}$, say j, and outputs $j+2$. As long as this guess is consistent with the data fed to M in subsequent stages, it repeats its hypothesis. Note that, by definition of $\mathcal{L}$, there may only occur inconsistencies with negative data. If an inconsistency is detected, say on i_{x+r} for some $r \in N$, then M again computes $I = \{k \mid b^k \in i_{x+r}^-\}$ as well as I's index j' in the enumeration $(I_j)_{j\in N}$. Moreover, it outputs $j'+2$. The latter guess is repeated as long as it is consistent. Otherwise, the construction is iterated.

Case 2: $w \in \{b\}^*$

Then M outputs 2. Next M proceeds as follows. As long as M only receives negative data over $\{a\}^*$ or positively marked strings over $\{b\}^*$, it repeats its guess. If in i_x appears the first time a string $(w,-) \in \{b\}^* \times \{+,-\}$, then it computes $I = \{k \mid b^k \in i_x^-\}$ and I's number in the enumeration $(I_j)_{j\in N}$, say j, and outputs $j+2$. Subsequently it behaves exactly as in case 1.

Note that there are no more cases to deal with, since $L \cap (\{a,b\}^* \setminus (\{a\}^* \cup \{b\}^*)) = \emptyset$ for all $L \in \mathcal{L}$. It can be straightforwardly shown that M infers $\mathcal{L}$ monotonically. We omit the details.

Claim 2: $\mathcal{L} \notin IT-INF$

Suppose the converse, i.e., there is an IIM inferring iteratively each $L \in \mathcal{L}$ on any informant for L. We construct a language $L = L(I)$ for some finite set I as well as an informant i' on which M does not identify L. This is done in two stages. First we consider M's behavior on some arbitrarily fixed informant i for L_1. Next we choose an appropriate set I_{x_1} as well as the corresponding language L_{x_1}. Then we investigate M's behavior on an informant $\tilde{i}$ for L_{x_1} that coincides on a sufficiently large initial segment with i. Finally, we add some appropriate chosen number to I_{x_1} yielding the desired I. The informant i' on which we fool M contains then the information concerning the added number at a place on which M ignores it. Hence, M is fooled. Formally the proof is performed as follows.

We say that M stabilizes itself on j_x at point x on an informant i iff $M_{x-1}(i) \neq j_x$, and $M(j_x, (w_{x+r}, b_{x+r})) = j_x$ for all $r \geq 0$. Since M is supposed to identify L_1 on i, it has to stabilize on a correct hypothesis j_{x_1} at some point x_1. Consequently, in particular we have

(A) $M(j_{x_1}, (w,b)) = j_{x_1}$ for all $(w,b) \in i$, $w \notin range(i_{x_1})$.

At this point we mainly use the fact that M works iteratively. Remember that $L_1 = \{a\}^*$. Consequently, from j_{x_1} the machine M cannot derive any information concerning $i_{x_1}^-$. Next we consider $I_{x_1} = \{n \mid b^n \in i_{x_1}^-\}$.

Case 1: $I_{x_1} = \emptyset$

Then we may choose any $x > x_1$ for which $I_x = \{n \mid b^n \in i_x^-\} \neq \emptyset$. Therefore we are, without loss of generality, in case 2.

Case 2: $I_{x_1} \neq \emptyset$

Consider an informant $\tilde{i}$ for $L_{x_1} = L(I_{x_1})$ satisfying $i_{x_1} = \tilde{i}_{x_1}$. By assumption M has to infer L_{x_1} on $\tilde{i}$, too. Consequently, there has to be an $x_2 > x_1$ at which M stabilizes on a correct hypothesis j_{x_2} for L_{x_1}, i.e., in particular we have:

(B) $M(j_{x_2}, (w,b)) = j_{x_2}$ for all $(w,b) \in \tilde{i}$, $w \notin range(\tilde{i}_{x_2})$.

Let $w = b^n$ with $w \notin range(\tilde{i}_{x_2})$ be arbitrarily fixed.

¿From (A) we immediately obtain that $M(j_{x_1}, (w,-)) = j_{x_1}$, since $(w,-)$ somewhere in i and $w \notin range(i_{x_1})$. Finally, let $I = I_{x_1} \cup \{n\}$. Obviously, $I \neq I_{x_1}$ and therefore we have $L_{x_1+2} \neq L := L(I)$. We define the wanted informant i' as follows: Let $i(z)$ denote the zth member of sequence i. We set:

$$i'(z) = \begin{cases} i(z) & , \ if \quad z \leq x_1 \\ (b^n, -) & , \ if \quad z = x_1 + 1 \\ \tilde{i}(z-1) & , \ if \quad z > x_1 + 1 \end{cases}$$

Therefore, in accordance with the latter observation and by construction we obtain that $j_{x_1} = M(j_{x_1}, \tilde{i}(x_1)) = M(j_{x_1}, i'(x_1)) = M(j_{x_1}, i'(x_1+1))$, and hence by (B) $M(j_{x_1}, i'(x_1+1+k)) = M(j_{x_1}, \tilde{i}(x_1+k))$ for all $k \geq 1$. Consequently, M on i' converges to j_{x_2}, a contradiction. This proves the lemma.

Lemma B. $IT-TXT \setminus MON-INF \neq \emptyset$

Proof (Lemma B). The lemma can be proved using the following indexed family $\mathcal{L}_{it}$: Let $L_1 := \{a\}^*$ and for $k > 1$ set $L_k := \{a^z \mid z < k\} \cup \{b^z \mid z \geq k\}$ as well as $L_{k,j} := \{a^z \mid z < k\} \cup \{b^z \mid k \leq z < j\} \cup \{a^z \mid z \geq j\} \cup \{c^j\}$ for all $k,j \in N$ with $k < j$, and $k,j \geq 1$. $\mathcal{L}_{it}$ is defined to be the collection of all L_k, $L_{k,j}$.

Claim A: $\mathcal{L}_{it} \notin MON-INF$

Suppose the converse, i.e., assume $\mathcal{L}_{it} \in MON-$

$INF(M)$ for some IIM M. Let i be any informant for $\{a\}^*$. Since $L_1 \in \mathcal{L}_{it}$ there must be an x such that $j_x = M(i_x)$ and $L(G_{j_x}) = L_1$. Next we successively enlarge i_x by $(b^z, +)$, where $z \geq y = max\{|w| \mid w \in i_x^+ \cup i_x^-\} + 1$. Consequently, all i_{x+k} are initial segments of an informant for L_y. Hence there must be a number k such that M on i_{x+k} outputs a grammar j_{x+k} being correct for L_y. But now we may enlarge i_{x+k} in a canonical manner to an informant i_{fool} for $L_{y,m}$ where $m = max\{|w| \mid w \in i_{x+k}^+ \cup i_{x+k}^-\} + 1$. It is easy to see that M either does not work monotonically on i_{fool} or it does not infer $L_{y,m}$. This proves the claim.

Claim B: $\mathcal{L}_{it} \in IT - TXT$
Let $L \in \mathcal{L}_{it}$ and let t be any text for L. The wanted IIM inferring $\mathcal{L}_{it}$ with respect to $\mathcal{L}_{it}$ is informally defined as follows. Initially M outputs 1. This guess is repeated until M receives the first time a string $s = b^z$. Since M actually gets its last guess j and a string, it can obviously check whether $j = 1$. Then M outputs z. As long as M's subsequent inputs are strings over $\{a\}$, the guess z remains unchanged. If $s = b^k$ is fed to M, it tests whether or not $b^k \in L_z$. In case it is, M outputs z. Otherwise, the new hypothesis is k. If M receives $s = c^m$ as input, it changes its mind to (z, m) or (k, m), respectively. Subsequently M behaves as follows. If the input is a string over $\{a\}$, it repeats its input guess. In case it receives a string b^l, it tests whether or not $b^l \in L_{z,m}$, where (z, m) is the hypothesis fed to M. If it is, the guess (z, m) is repeated. Otherwise, it changes its mind to (l, m), a.s.o.

A straightforward argumentation shows that $\mathcal{L}_{it} \in IT - TXT(M)$. We omit the details. This proves Lemma B.

Finally, by Lemma A we immediately get assertion (1), since $WMON - INF = LIM - INF$ and $IT - INF \subseteq WMON - INF$ by definition. Moreover, since obviously $MON - IT - INF \subseteq IT - INF$, Lemma A implies assertion (4), and by an analogous argument (3), too. The remaining assertion (2) is obtained by Lemma A and Lemma B. Hence the Theorem is proved.

q.e.d.

The relation between $SMON - IT - INF$ and $SMON - INF$ is not completely solved. Next we deal with iterative learning combined with monotonicity requirements on positive data. Moreover, we also ask whether one can trade information presentation versus monotonicity requirements. The next theorem summarizes the results obtained.

Theorem 6.

(1) $IT - TXT \subset IT - INF$
(2) $MON - IT - TXT \subset MON - TXT$
(3) $WMON - IT - TXT \subset WMON - TXT$
(4) $MON - TXT \# IT - TXT$
(5) $WMON - TXT \# IT - TXT$

Proof. The first part of assertion (1), i.e., $IT - TXT \subseteq IT - INF$ is obvious. The proper inclusion is shown using the following indexed family $\mathcal{L}_{inf} = (L_j)_{j \in N}$ over the alphabet $\{a\}$, where $L_1 = \{a\}^*$ and $L_j = \{a^z \mid z \leq j\}$ for $j > 1$.

Claim 1: $\mathcal{L}_{inf} \notin LIM - TXT$
Suppose the converse. Then by Angluin's characterization of $LIM - TXT$ each L_j should have a finite tell-tale. However, a straightforward argument directly yields that L_1 cannot possess a finite tell-tale. This proves the claim.

Since by definition $IT - TXT \subseteq LIM - TXT$, we have $\mathcal{L}_{inf} \notin IT - TXT$. After a bit of reflection one sees that $\mathcal{L}_{inf} \in IT - INF$. This proves assertion (1).

Claim 2: $IT - TXT \setminus WMON - TXT \neq \emptyset$
This claim may be proved using the indexed family Angluin (1980) has used to obtain $CONSERVATIVE\text{-}TXT \subset LIM - TXT$. We omit the details.

Claim 3: $MON - TXT \setminus IT - TXT \neq \emptyset$
We define an indexed family $\mathcal{L}_{mon}$ over the alphabet $\{a, b, c\}$ as follows. Let $L_1 = \{a\}^*$ and $L_{k,n} = \{a^z \mid z \leq k\} \cup \{a^z \mid z \geq n\} \cup \{b^k, b^n\} \cup \{c\}$ for all $k, n \in N$, $k, n > 1$ and $k+2 < n$. Finally, we set $L_{k,n,m} = L_{k,n} \setminus \{c\} \cup \{a^m\}$ for all $k, n \in N$ as above and $k < m < n$. Then define $\mathcal{L}_{mon}$ to be the collection of all L_1, $L_{k,n}$ and $L_{k,n,m}$. It is not hard to prove that $\mathcal{L}_{mon} \in MON - TXT \setminus IT - TXT$. This proves claim 3.

Obviously, Claim 1, 2 and 3 directly yield the remaining assertions.

q.e.d.

Finally, we have been very surprised in obtaining the following theorem. Theorem 7 gives another characterization of strong–monotonic inference from positive data.

Theorem 7.

(1) $SMON - TXT \subset IT - TXT$

(2) $SMON - IT - TXT = SMON - TXT$

Proof. First observe that we have already proved $IT - TXT \setminus SMON - TXT \neq \emptyset$, since $SMON - TXT \subset MON - INF$. Therefore it suffices to show $SMON - TXT \subseteq IT - TXT$. Let $\mathcal{L} \in SMON - TXT$, i.e., there are an IIM M as well as a space $\mathcal{G} = (G_j)_{j \in n}$ of hypotheses such that, without loss of generality, $\mathcal{L} \in SMON - CONS - TXT(M)$ with respect to $\mathcal{G}$ (cf. Lange and Zeugmann (1992)). Moreover, M may even supposed to output always an hypotheses, since we are allowed to output guesses that not necessarily describe a language contained in $\mathcal{L}$. We proceed in defining an IIM $\tilde{M}$ as well as a space $\tilde{\mathcal{G}} = (\tilde{G}_j)_{j \in N}$ of hypotheses. Thereafter we show that $\tilde{M}$ iteratively infers $\mathcal{L}$ with respect to $\tilde{\mathcal{G}}$. The desired space of hypotheses $\tilde{\mathcal{G}}$ is obtained from $\mathcal{G}$ by enumerating its closure with respect to finite unions. $\tilde{M}$ is defined in stages, where stage k conceptually describes $\tilde{M}_k$. Let $L \in \mathcal{L}$ and let t be any arbitrarily fixed text for L.

Stage 0: Let $t_1 = s_1$. Compute $j_1 := M(s_1)$. Output j_1, and goto stage 1.

Stage k: $\tilde{M}$ receives as input j_{k-1} and the kth element s_k of t. Test whether or not $s_k \in L(\tilde{G}_{j_{k-1}})$.

Case 1: $s_k \in L(\tilde{G}_{j_{k-1}})$.
Set $j_k = j_{k-1}$, output it, and goto stage $k+1$.

Case 2: $s_k \notin L(\tilde{G}_{j_{k-1}})$.
Test for all strings $w \in \Sigma^*$ with $|w| \leq |s_k|$ whether or not $w \in L(\tilde{G}_{j_{k-1}})$. Let $\tilde{w}_1, ..., \tilde{w}_l$ be the strings successfully passing the test plus s_k written in lexicographical order. Compute $\tilde{j}_k := M(\tilde{w}_1, ..., \tilde{w}_l)$.
(* By construction we have $\{\tilde{w}_1, ..., \tilde{w}_l\} \subseteq L(G_{\tilde{j}_k})$, since M works consistently *)
Finally, compute a canonical number j_k for $L(\tilde{G}_{j_{k-1}}) \cup L(G_{\tilde{j}_k})$. Output j_k and goto stage $k+1$.

We continue in proving $\mathcal{L} \in IT-TXT(\tilde{M})$ with respect to $\tilde{\mathcal{G}}$. First, $\tilde{M}$ works iteratively by definition. Next, for all guesses j_k output by $\tilde{M}$ we obtain $L(\tilde{G}_{j_k})$, since M works strong–monotonically. Moreover, after stage k it always holds that $t_k^+ \subseteq L(\tilde{G}_{j_k})$. By construction we additionally get $L(\tilde{G}_{j_k}) \subseteq L(\tilde{G}_{j_{k+r}})$ for any $r \in N$. Finally, if $\tilde{M}$ produces a guess j_k at all such that $L = L(\tilde{G}_{j_k})$, then this guess is maintained in any subsequent stage. Hence showing that $\tilde{M}$ converges on t reduces to proving that $\tilde{M}$ on t has to output a correct hypothesis in some stage.

Claim: $\tilde{M}$ converges on t.
We distinguish the following two cases.

Case 1: L is finite.
Since L is finite, by the observations made above, we immediately obtain that there are only finitely many stages at which $\tilde{M}$ can detect an inconsistency. Moreover, as we pointed out, each string $s \in L$ may force $\tilde{M}$ at most once to perform a mind change. Finally, since $\tilde{M}$ only outputs guesses that are contained in L, it sometime reaches a correct hypothesis. This proves the claim.

Case 2: L is infinite
Suppose the converse, i.e., $\tilde{M}$ changes its mind infinitely often. Consequently, since any mind change is forced by a detected inconsistency, there should be infinitely many *different* strings $s \in L$ disproving $\tilde{M}$'s current guess. However, this cannot happen as the following argumentation shows. Let t^{ord} be L's lexicographically ordered text. Since M has to converge on t^{ord}, too, there is a $z \in N$ such that $M(t_z^{ord}) = j$ and $L = L(G_j)$. Moreover, taking into account that M works strong–monotonically, we additionally obtain: Any hypothesis $h = M(t_z^{ord}\sigma)$ do satisfy $L = L(G_h)$ for any sequence σ of strings from L. Furthermore, since t is a text for L, there exists an $x \in N$ such that $range(t_z^{ord}) \subseteq t_x^+$. Finally, as we have seen above, each string $s \in L$ may force $\tilde{M}$ at most once to perform a mind change. Consequently, at least in stage x the machine $\tilde{M}$ outputs a guess j_x such that $t_x^+ \subseteq L(\tilde{G}_{j_x})$. However, in accordance with our assumption $\tilde{M}$ receives in some subsequent stage, say $x+r$, a string s such that $s \notin L(\tilde{G}_{j_{x+r-1}})$. As mentioned above, $L(\tilde{G}_{j_x}) \subseteq L(\tilde{G}_{j_{x+r-1}})$. Thus $s \notin L(\tilde{G}_{j_x})$. Therefore $|s| \geq max\{|w| \mid w \in range(t_z^{ord})\}$. That means, in stage $x+r$ the machine $\tilde{M}$ computes a sequence $\tilde{w}_1, ..., \tilde{w}_l$ that has t_z^{ord} as prefix. Now $\tilde{M}$ computes $M(\tilde{w}_1, ..., \tilde{w}_l) = M(t_z^{ord}\sigma) = h$, and as we have seen above, $L = L(G_h)$. Hence $\tilde{M}$'s output at the end of stage $x+r$ is a correct guess. This proves the claim as well as assertion (1).

Moreover, a closer look to the proof presented above yields that we have actually shown $SMON-TXT \subseteq SMON-IT-TXT$. Since the converse inclusion is obvious, the latter observation implies assertion (2). Hence the theorem is proved.

q.e.d.

5 Conclusions and Open Problems

We have characterized strong–monotonic, monotonic as well as weak monotonic learning from positive data. In particular, the characterization of $WMON-TXT$ solved the problem of how to characterize inference algorithms that avoid overgeneralization.

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 be eventually 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 (1992) we have shown that there is an indexed family of recursive languages such that set inclusion is uniformly decidable but which is not *monotonically* inferrable, even on *informant*.

However, several problems remained open. One of the most intriguing questions is whether or not all types of 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 and 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 $SMON-TXT \subset MONR-TXT$ (cf. Lange and Zeugmann (1992)).

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

Finally, we have dealt with the problem to perform strong–monotonic, monotonic and weak–monotonic inference with iteratively working $IIMs$. As it turned out, strong–monotonic inference from positive data may always be done with iteratively learning devices without decreasing the learning capabilities. On the other hand, monotonic and weak–monotonic inference with iterative $IIMs$ is less powerful than with ordinary machines. However, it remained open whether strong–monotonic inference on informant can be performed by iteratively working machines without limiting the learning power.

Acknowledgement

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

6 References

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

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

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

[4] Barzdin, Ya.M., (1974), Inductive Inference of Automata, Functions and Programs, Proc. International Congress of Math., Vancouver, pp. 455 - 460

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

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

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

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

[9] Freivalds, R., Kinber, E. B. and R. Wiehagen, (1992), Convergently versus Divergently Incorrect Hypotheses in Inductive Inference, GOSLER Report 02/92, January 1992, Fachbereich Mathematik und Informatik, TH Leipzig

[10] Fulk, M.,(1990), Prudence and other Restrictions in Formal Language Learning, Information and Computation 85, 1 - 11

[11] Gold, M.E., (1967), Language Identification in the Limit, Information and Control 10, 447 - 474

[12] Jain, S. and A. Sharma, (1989), Recursion Theoretic Characterizations of Language Learning, The University of Rochester, Dept. of Computer Science, TR 281

[13] Jantke, K.P., (1991A), Monotonic and Non–monotonic Inductive Inference, New Generation Computing 8, 349 - 360

[14] Jantke, K.P., (1991B), Monotonic and Non–monotonic Inductive Inference of Functions and Patterns, Proc. First International Workshop on Nonmonotonic and Inductive Logics, December 1990, Karlsruhe, J.Dix, K.P. Jantke and P.H. Schmitt (Eds.), Lecture Notes in Artificial Intelligence 543, pp. 161 - 177, Springer-Verlag

[15] Kapur, S.,(1992), Computational Learning of Languages, Department of Computer Science, Cornell University, Ithaca, New York, Ph.D Thesis, TR 91-1234, January 1992

[16] Kapur, S. and G. Bilardi, (1992), Language Learning without Overgeneralzation, Proc. 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 577, pp. 245 - 256, Springer-Verlag

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

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

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

[20] 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

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

[22] Porat, S. and J.A. Feldman, (1988), Learning Automata from Ordered Examples, in Proc. First Workshop on Computational Learning Theory, D. Haussler and L. Pitt (Eds.), pp. 386 - 396, Morgan Kaufmann Publ.

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

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

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

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

[27] Wiehagen, R., (1992), Personal Communication