Language Learning from Texts: Mind Changes, Limited Memory and Monotonicity

(Extended Abstract) *

Efim Kinber †
University of Delaware

Frank Stephan ‡
Universität Karlsruhe

Abstract

The paper explores language learning in the limit under various constraints on the number of mind-changes, memory, and monotonicity. We define language learning with limited (long term) memory and prove that learning with limited memory is exactly the same as learning via set driven machines (when the order of the input string is not taken into account). Further we show that every language learnable via a set driven machine is learnable via a conservative machine (making only justifiable mindchanges).

We get a variety of separation results for learning with bounded number of mindchanges or limited memory under restrictions on monotonicity. Many separation results have a variant: If a criterion $\mathcal{A}$ can be separated from $\mathcal{B}$, then often it is possible to find a family $\mathcal{L}$ of languages such that $\mathcal{L}$ is $\mathcal{A}$ and $\mathcal{B}$ learnable, but while it is possible to restrict the number of mindchanges or long term memory on criterion $\mathcal{A}$, this is impossible for $\mathcal{B}$.

1 Introduction

Learning languages from texts has become a subject of intensive research within recent years. One of the central problems in this area is: how do various restrictions on the behaviour of a learner limit the learning abilities? We consider three types of restrictions: monotonicity-requirements, limitations on number of mindchanges and on memory. But we do nowhere restrict the families to be learned; i.e., we learn r.e. indices of languages from arbitrary families of r.e. sets unlike Angluin [1], who considered learning r.e. families of recursive sets under similar constraints.

Our first restriction is on the amount of memory used by a learner. Already [5, 19] considered, among many other models, some, where the learner had access only to the most recent input and most recent conjectures. Freivalds, Kinber and Smith [6] created a model for distinguishing between short term memory, which is cleared before reading a new word from the input text, and long term memory, which keeps information on previous words and results, but is limited in size. Kinber [12] applied this concept to learning classes of r.e. languages and obtained some first results. Our major result for this model is that families learnable with limited long term memory are exactly those learnable via a set driven machine (whose behaviour does not depend on the input order). This result sheds a new light on the nature of set driven learning. For instance, it has enabled us to show that any set driven learnable family is conservatively learnable (when the learner can change his mind only if a new evidence has appeared) with linear memory, as well as to show that there are conservatively learnable families which can not be learned by a set driven machine.

Conservativeness fits into the various monotonicity constraints which were introduced by Jantke [10] and Kapur [11]. These constraints model learning by generalization and by specialization. The intention of these two notions is that the learner, being fed more and more positive examples of the language, produces better and better generalizations (specializations, respectively). In the strongest interpretation, the learner has to infer a sequence of hypotheses describing a growing (descending, respectively) chain of languages, i.e., $L_i \subseteq L_j$ ($L_i \supseteq L_j$, respectively) if the learner guessed L_j later than L_i. Jantke, Kapur, Lange and Zeugmann in various publications [10, 11, 13, 17] defined and explored Several natural approaches to monotonicity. Many non-inclusion facts from this field are sharpened as follows: a family is learnable with a small memory in one sense and non-learnable in a stronger sense.

†Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716, U.S.A., email ⟨kinber@cis.udel.edu⟩.

‡Institut für Logik, Komplexität und Deduktionssysteme, Universität Karlsruhe, 76128 Karlsruhe, Germany, email ⟨fstephan@ira.uka.de⟩. Supported by the Deutsche Forschungsgemeinschaft (DFG) grant Me 672/4-2.

Third, we use the number of mindchanges a learner makes on a text as a measure of learning complexity. Learning with restrictions on the number of mindchanges was widely explored in many works, e.g. in [13, 18]. We found that if some monotonicity requirement $\mathcal{A}$ implies $\mathcal{B}$, then any family of languages $\mathcal{L}$ which is $\mathcal{A}$ inferable with k mindchanges is also $\mathcal{B}$ inferable with k mindchanges. On the other hand, if $\mathcal{A}$ does not imply $\mathcal{B}$, then there is a family of languages learnable according to $\mathcal{A}$ with 2 mind changes which still fails to be learnable according to $\mathcal{B}$.

2 The Learning Requirements

We consider the Gold-style [4, 7, 19] formal language learning model. An algorithmic learning device, being fed the sequence of strings s in the target language L and symbols # (representing pauses in the presentation of data), produces a sequence of hypotheses $H_1, H_2, \ldots$ such that the limit of this sequence is a program for the target language. More formally, let Σ denote a fixed finite alphabet of symbols; often we use $\Sigma = \{0,1\}$ or $\Sigma = \{0,1,*\}$. In languages natural numbers are always identified with their binary representation, e.g. 5 means 101.

Any subset $L \subseteq \Sigma^*$ is called a language. $\overline{L}$ denotes the complement $\Sigma^* - L$ of L. We consider only r.e. languages, W_e is the e-th language according to some fixed acceptable numbering of all r.e. languages. Let $\# \notin \Sigma$. An infinite sequence $T \in (L \cup \#)^\infty$ is called a text for L if every word in L appears at least once in T. The range of a text or a finite initial segment $\sigma \preceq T$ is the set of all words unequal # appearing in T (σ). Furthermore, let $\sqsubset$ denote a recursive linear ordering of all finite texts in $(\Sigma^* \cup \{\#\})^*$ such that $|\sigma| < |\tau| \Rightarrow \sigma \sqsubset \tau$.

Following Gold [7], we define an Inductive Inference Machine (IIM) to be an algorithmic device which works as follows: it takes larger and larger initial segments $\sigma \preceq T$ and either requires the next input string w, or it first outputs a hypothesis, i.e., a number e to guess the set W_e, and then it requests the next input string. Throughout this paper, we always consider learning from text and never from informant; therefore we do not indicate explicitly in the names of the learning criteria that they are TEXT learning criteria.

Definition 2.1 Let $\mathcal{L}$ be a family of languages and $L \in \mathcal{L}$. An IIM M LIM identifies L on a text T iff there is some index e such that $L = W_e$, $M(\sigma) = e$ for some $\sigma \preceq T$ and $M(\tau) \in \{e, ?\}$ for all $\tau \succeq \sigma$ with $\tau \prec T$. Here the symbol "?" denotes that M does not want to make a guess. "?" is needed in the special cases of limited long term memory (if M cannot remember its last guess but does not want to make a new one) and bounded mindchanges (e.g., if the first guess must be correct, but M has not seen sufficient information to make up its mind). Moreover, an IIM M LIM infers L iff it LIM identifies L on every text for L. For any $k \in N$, we say that an IIM M identifies L with k mindchanges, if for any text T for L, M outputs at most $k{+}1$ different guesses $e_0, \ldots, e_k$ and never returns to an old e_i after once guessing e_{i+1}.

Note that we do not require any properties of $\mathcal{L}$ and the guesses e such as $\mathcal{L}$ being an r.e. family of sets, or each e being a characteristic index.

Definition 2.2 Following Freivalds, Kinber and Smith [6], we assume that every IIM M has two types of memory: long term memory and short term memory. M uses its long term memory to remember any information that can be useful in later stages of inference; for instance M memorizes portions of the input it has seen or prior conjectures. The short term memory is potentially unlimited and is annihilated every time the IIM either outputs a new conjecture or begins to read a new word in the input. The short term memory clearing is done automatically and takes one time step. Separation of the short term memory from the long term memory is very useful (and proved to be very fruitful in [6]) to ensure an accurate accounting of the real long term memory needed for learning the unknown language. The limitation on the size of the long term memory after reading the input σ is always a function of the size of $range(\sigma)$ — therefore if a finite set is learned, then the long term memory is limited during the whole inference process by a constant depending on this set.

Definition 2.3 Informally speaking, an IIM learns monotonically if it produces better and better generalizations. However, monotonicity and dual monotonicity can be defined mathematically in various ways. Here we follow [10, 11, 13, 17]. Let the IIM M identify an language L from text. On this inference process M is said to satisfy the additional requirement

SMON (strongly-monotonic)
  iff $W_e \subseteq W_{e'}$
SMON$^{\mathrm{d}}$ (dual strongly-monotonic)
  iff $W_e \supseteq W_{e'}$
MON (monotonic)
  iff $W_e \cap L \subseteq W_{e'} \cap L$
MON$^{\mathrm{d}}$ (dual monotonic)
  iff $W_e \cup L \supseteq W_{e'} \cup L$
CONV (conservative)
  iff $range(\sigma') \subseteq W_e \Rightarrow e = e'$
WMON (weakly-monotonic)
  iff $range(\sigma') \subseteq W_e \Rightarrow W_e \subseteq W_{e'}$
WMON$^{\mathrm{d}}$ (dual weakly-monotonic)
  iff $range(\sigma') \subseteq W_e \Rightarrow W_e \supseteq W_{e'}$

for all guesses $e = M(\sigma)$ and $e' = M(\sigma')$ with $\sigma \preceq \sigma'$ and $\sigma, \sigma' \in (L \cup \#)^*$. The requirements SMON and SMON$^{\mathrm{d}}$ are straightforward and very strong. But they are also of limited power: If a SMON learner erroneously adds a word to a hypothesis, it cannot remove this word from the target language description. The requirements MON and MON$^{\mathrm{d}}$ are designed to overcome this difficulty while keeping as much of the original requirements as possible. CONV inference permits only reasonable mindchanges: the learner may only make a new conjecture if the old one is definitely inconsistent with the data seen so far. WMON and $\mathrm{WMON}^{\mathrm{d}}$ are variants, which try to integrate the ideas of conservativism and monotonicity.

SMON:f, $\mathrm{SMON}^{\mathrm{d}}$:$f$, ... denote the combinations of limited memory and the monotonicity requirements. E.g., SMON:f denotes that an IIM, whose memory is limited by f, infers each set L by an ascending sequence of guesses $W_{e_1} \subseteq W_{e_2} \subseteq \ldots \subseteq W_{e_k} = L$.

3 Technical Summary

We study the connection between different requirements of learning, in particular, between limitations on the usage of long term memory, monotonicity requirements and bounds on the number of mindchanges:

Limited memory and set driven inference:
A class of languages is learnable with some bound on the long term memory iff it is learnable via a set driven IIM. Any set driven IIM can be made conservative, but there are classes of languages learnable via a conservative IIM, but not learnable via a set driven IIM.

Limited memory hierarchy:
If two functions $f < g < \mathrm{id}$, then there is a class of languages which can be learned strongly monotonic using the memory bound g but which is not LIM:f learnable.

Limited memory and monotonicity:
Between the monotonicity-requirements SMON, $\mathrm{SMON}^{\mathrm{d}}$, MON, $\mathrm{MON}^{\mathrm{d}}$, LIM only the trivial inclusions hold, which also preserve bounds on the long term memory. Every non-inclusion $\mathcal{A} \not\to \mathcal{B}$ is witnessed via a set learnable using only a constant amount of memory. Also some $\mathcal{B}$:id learnable set can be learned with constant amount of memory under requirement $\mathcal{A}$ while requirement $\mathcal{B}$ does not permit any more restrictive bound on the use of long term memory.

Constant long term memory:
If a class of sets is learnable via a constant amount of memory, then it is also learnable with a constant bound on the number of mindchanges. This transition preserves the monotonicity requirements. But the other way round does not hold, i.e., there is a class learnable with at most one mindchange which can not be learned via any set driven machine.

Monotonicity requirements and bounded number of mindchanges:
WMON $\to$ CONV and LIM $\to$ $\mathrm{WMON}^{\mathrm{d}}$ are the only non-trivial inclusions; on the other hand the non-inclusion MON $\not\to$ WMON differs from results in more restricted contexts. All inclusions preserve the bounds on the number of mindchanges while the non-inclusions are already witnessed by classes of sets learnable via only two mindchanges.

4 The Limited Memory Hierarchy

This section analyzes the hierarchy of language classes learnable with limited long term memory. The main results are:

- The classes LIM:f of all f with $(\forall x)[f(x) \geq \mathrm{id}(x)]$ coincide. Here id denotes the amount of memory to store $range(\sigma)$ of any text σ. Our machine model is chosen such that $\mathrm{id}(x) = x$, for all x: Given a function f the IIM has the right to store a finite set $\{w_1, \ldots, w_n\}$ of strings such that $|w_1| + \ldots + |w_n| \leq f(|v_1| + \ldots + |v_m|)$ and $n \leq m$ where $\{v_1, \ldots, v_m\}$ is the range of the input seen so far.
- It turns out that the class LIM:id is just the class of languages learnable via a set driven IIM. This class coincides with CONV:id, but it is properly contained in the class of all CONV learnable languages.
- There is some $\mathcal{L}$ which is LIM:f learnable but not LIM:g iff there is a x with $g(x) < f(x) \leq x$.
- The following diagram illustrates the results graphically:

$$\begin{tikzcd}
& \mathrm{LIM} : c \arrow[dl] \arrow[dr] & \\
\mathrm{LIM} : f \arrow[dr] & & \mathrm{LIM} : g \arrow[dl] \\
\mathrm{CONV} : \mathrm{id} \arrow[r, leftrightarrow] & \mathrm{LIM} : \mathrm{id} \arrow[r, leftrightarrow] \arrow[d] & \mathrm{SET\ DRIVEN} \\
& \mathrm{CONV} \arrow[d] & \\
& \mathrm{LIM} &
\end{tikzcd}$$

  where f and g are incomparable functions with $c \leq f, g \leq \mathrm{id}$.

Wexler and Culicover [5, Section 2.2] defined that an IIM M is set driven if $M(\sigma) = M(\tau)$ whenever $range(\sigma) = range(\tau)$. Dealing with a set driven IIM M it is often easier to write $M(range(\sigma))$ instead of $M(\sigma)$.

Kinber [12] asked, whether there is some LIM learnable family which is not LIM:f learnable for any (arbitrary fast growing) f. Gisela Schäfer [19, Proposition 4.4.2A] showed that there is a LIM learnable family $\mathcal{L}$ which is not inferred via any set driven IIM. By Theorem 4.1 (b)$\Rightarrow$(a), Schäfer's $\mathcal{L}$ is actually not LIM:f learnable for any f since $\mathcal{L}$ otherwise would also be learnable via a set driven IIM. So $\mathcal{L}$ witnesses a positive answer of Kinber's question.

Theorem 4.1 *The following statements are equivalent for a family $\mathcal{L}$:*
(a) *Some set driven* IIM *infers $\mathcal{L}$.*
(b) *$\mathcal{L}$ is* LIM:g *learnable for some g.*
(c) *$\mathcal{L}$ is* CONV:id *learnable.*

Proof: We show (b)$\Rightarrow$(a) and (a)$\Rightarrow$(c); the direction (c)$\Rightarrow$(b) is obvious.

Proof of (b)⇒(a): An IIM LIM:g learning some family $\mathcal{L}$ is given by two partial recursive functions m and M such that on long term memory x, $m(x,w)$ is the new long term memory after reading the input word w and $M(x,w)$ is its guess (where $M(x,w) = ?$ stands for "no new guess"). Further let $\tilde{m}(\lambda) = \lambda$ and $\tilde{m}(\sigma w) = m(\tilde{m}(\sigma), w)$; here the long term memory is considered as a string initialized to λ. So $\tilde{m}(\sigma)$ is the content of the long term memory after reading σ. The guess after reading the input σw is $M(\tilde{m}(\sigma), w)$. M, m and $\tilde{m}$ are partial recursive functions such that $(L \cup \{\#\})^* \subseteq dom(\tilde{m})$ and $\tilde{m}((L\cup\{\#\})^*) \times (L \cup \{\#\}) \subseteq dom(m), dom(M)$ for any $L \in \mathcal{L}$.

Let $L \in \mathcal{L}$, $W \subseteq L$ and n be the size of the finite set W. $|\tilde{m}(\sigma)| \leq g(n)$ on every input $\sigma \in (W \cup \{\#\})^*$, thus the set $X(W) = \{\tilde{m}(\sigma) : \sigma \in (W \cup \{\#\})^*\}$ is finite and r.e.; furthermore $m(x,w) \in X(W)$ for all $x \in X(W)$ and $w \in W$. So one enumerates $X(W)$ until the first stage t such that

(I) $m_t(x,w)\downarrow \in X_t(W)$ for all $x \in X_t(W)$ and $w \in W \cup \{\#\}$;
(II) $M_t(x,w)\uparrow$ for all $x \in X_t(W)$ and $w \in W \cup \{\#\}$;
(III) $\lambda \in X_t(W)$.

Since $X(W)$ is finite, this stage will be reached. Furthermore, whenever at some t the conditions (I), (II) and (III) are satisfied, no new element will enter $X(W)$; i.e., $X(W) = X_t(W)$. Now M_t is the output-function and m_t the transition-function for a finite automaton, whose states are the elements of $X(W)$.

On the finite automaton, it is possible to check whether σw is a locking sequence for W, i.e., whether $range(\sigma w) \subseteq W$, $(\exists e \neq ?)[M(\tilde{m}(\sigma), w) = e]$ and $M(\tilde{m}(\sigma w \tau), v) \in \{?, e\}$ for all $\tau v \in (W \cup \{\#\})^+$. If there is a locking sequence, then there is one of length at most $(|X(W)|+1)$, thus it is decidable whether there is a locking sequence or not. A set driven IIM N for $\mathcal{L}$ is given by

$$N(W) = \begin{cases} M(\tilde{m}(\sigma), w) & \text{if } \sigma w \text{ is a locking sequence for } W \text{ and no } \tau v \sqsubset \sigma w \text{ is a locking sequence for } W; \\ ? & \text{otherwise, i.e., there is no locking sequence for } W. \end{cases}$$

If L is finite, then N will output the correct value for $W = L$. If L is infinite, then there is a locking sequence. Let σw be the $\sqsubset$-first locking sequence. For all sequences $\tau v \sqsubset \sigma w$ with $e' = M(\tilde{m}(\tau), v)$ there is some extension $\eta u \succ \tau v$ such that $\eta u \in (L \cup \{\#\})^*$ and $M(\tilde{m}(\eta), u) \notin \{?, e'\}$. Thus they are not considered for any $W \supseteq range(\eta u)$ and so N converges to an index of L for all sufficiently large subsets of L.

Proof of (a)⇒(c): This is done in two stages. First M is replaced by a new IIM N such that N never returns to old output, i.e., whenever $N(U) \neq N(V)$ for finite sets U, W with $U \subseteq W$ then $N(W) \neq N(U)$ for all finite supersets W of V. In a second stage a direct modification s of the output of N is defined such that N' given by $N'(W) = s(N(W))$ is a conservative IIM for the class $\mathcal{L}$ inferred by M.

For the first step let p be an injective recursive padding function such that $W_{p(e,U,\mathcal{F})} = W_e$ for every index e, every finite set U and every finite class $\mathcal{F}$ of finite sets. Let $U(W)$ be the $\sqsubset$-first set $U \subseteq W$ with $M(V) = M(W)$ for all $V \in [U, W]$ and let $\mathcal{F}(W)$ contain all $F \sqsubset U(W)$ with $F \subseteq W$ and $M(F) \neq M(V)$ for some $V \in [F, U]$. Now let $N(W) = p(M(W), U(W), \mathcal{F}(W))$.

Obviously N is recursive and N is defined on input W whenever $M(V)$ is defined for all $V \subseteq W$. If L is finite then $N(L)$ computes an index of the set $W_{M(L)}$ and thus N infers every finite set $L \in \mathcal{L}$. If L is infinite then there is an index e of L and a $\sqsubset$-first finite set $U \subseteq L$ with $M(W) = e$ for all $W \in [U, L]$. Now $U(W) \sqsubseteq U$ for all $W \in [U, L]$. Note that if $V \subseteq W$ and $U(V) \subseteq U(W)$ then $\mathcal{F}(V) \subseteq \mathcal{F}(W)$. Therefore there is some $W \in [U, L]$ with $U(V) = U$ and $\mathcal{F}(V) = \mathcal{F}$ for all $V \in [W, L]$. Thus $N(V) = p(e, U, \mathcal{F})$ for all $V \in [W, L]$ and N infers also every infinite $L \in \mathcal{L}$.

So N also is an set driven IIM for $\mathcal{L}$. Now let $V_1 \subseteq V_2 \subseteq V_3$ be three finite sets. Assume that $N(V_3) = N(V_1) = p(M(V_1), U(V_1), \mathcal{F}(V_1))$, i.e., $M(V_1) = M(V_3)$, $U(V_1) = U(V_3)$ and $\mathcal{F}(V_1) = \mathcal{F}(V_3)$. First $M(V_2) = M(V_1)$ holds since otherwise $U(V_3) \not\subseteq V_1$. Second $U(V_2) \sqsubseteq U(V_1)$. $U(V_2) \notin \mathcal{F}(V_1)$, since otherwise $U(V_2) \subseteq V_1$ and there would be $V \in [U(V_2), V_1] \subseteq [U(V_2), V_2]$ such that $M(U(V_2)) \neq M(V)$ in contradiction to the choice of $U(V_2)$. From $U(V_2) \notin \mathcal{F}(V_3)$ and $U(V_2) \sqsubseteq U(V_3)$ follows $U(V_2) = U(V_3)$. Third from $U(V_1) = U(V_2) = U(V_3)$ and $V_1 \subseteq V_2 \subseteq V_3$ it follows that $\mathcal{F}(V_1) \subseteq \mathcal{F}(V_2) \subseteq \mathcal{F}(V_3)$ and from $\mathcal{F}(V_1) = \mathcal{F}(V_3)$, equality follows. So $N(V_2) = N(V_1)$. In other words N never returns to an old guess.

So N satisfies all requirments and the first step is complete. For the second step let

$$W_{s(e)} = \begin{cases} W_{e,t-1} & \text{for the first } t > 0 \text{ such that either } (\exists U \subseteq W_{e,t})[N(U)\uparrow] \\ & \text{or } (\exists U \subseteq W_{e,t})(\exists V \in [U, W_{e,t}]) [N(U) = e \wedge N(V) \neq e]; \\ W_e & \text{if there is no such } t. \end{cases}$$

where $W_{e,t}$ denotes the set of elements of W_e enumerated during the first t steps of a canonical uniform enumeration of all r.e. sets W_e; w.l.o.g. $W_{e,0} = \emptyset$. Now whenever $N(U) \neq N(V)$ and $U \subseteq V$ then $V \not\subseteq W_{N(U)}$, thus the inference process is conservative. Let $L \in \mathcal{L}$. Then there is some e with $L = W_e$ and some finite set $W \subseteq W_e$ with $N(V) = e$ for all $V \in [W, W_e]$. Assume by the way of contradiction that $W_e \neq W_{s(e)}$. Either there is some finite $V \subseteq W_e$ with $N(V)\uparrow$. This contradicts $L \in \mathcal{L}$. Or there are finite sets U, V with $N(U) = e$, $N(V) \neq e$ and $U \subseteq V \subseteq L$. But $N(V \cup W) = e$ and $W \subseteq L$, thus N returns to an old value in contradiction to the construction of N. So both cases fail and $W_{s(e)} = W_e$. N' given by $N'(W) = s(N(W))$ is a CONV:id IIM for L. ∎

Dana Angluin [1] and Gisela Schäfer [19, Proposition 4.4.2A] give an example for a language which is LIM learnable but cannot be learned either via a conservative or via a set driven IIM. The next result shows that there is also a language $\mathcal{L}$ which is conservative learnable but not via a set driven IIM.

Theorem 4.2 *There is a conservatively learnable family $\mathcal{L}$ which is not learnable via a set driven* IIM*; in particular $\mathcal{L}$ is not* LIM:*g learnable for any g.*

Proof: Let ψ be a partial recursive $\{0,1\}$-valued function which has no total recursive extension. Now let $U = \{i{*}\psi(i) : i \in dom(\psi)\}$, U has a recursive enumeration $\{U_s\}_{s\in\omega}$. A finite set V is incompatible with U_s iff $i{*}0, i{*}1 \in U_s \cup V$ for some i. V is incompatible with U iff V is incompatible with U_s for some s.

Let $\mathcal{L}$ consist of U and all finite sets V incompatible with U. The following IIM M infers the family $\mathcal{L}$ conservatively:

$$W_{M(\sigma)} = \begin{cases} U & \text{if } range(\sigma) \text{ is compatible with } U_{|\sigma|}; \\ range(\sigma) & \text{if } range(\sigma) \text{ is incompatible with } U_{|\sigma|}. \end{cases}$$

M is conservative, since the first mindchange from U to $range(\sigma)$ occurs only if $range(\sigma)$ is incompatible to U and thus $range(\sigma) \not\subseteq U$. All further guesses are canonical indices of finite sets, so that the conservativeness is not violated by any further mindchange. Also M infers U and all finite sets V incompatible to U.

Assume now that a set driven IIM N infers $\mathcal{L}$. Then there is a locking set $W \subseteq U$ such that $N(W') = N(W)$ for all finite sets W' with $W \subseteq W' \subseteq U$. Now define a recursive function f as follows:

$$f(i) = \begin{cases} 0 & \text{if } N(W \cup \{i{*}0\}) = N(W); \\ 1 & \text{otherwise.} \end{cases}$$

The function f is total and recursive since N is. If $\psi(i)\downarrow = 0$ then $i{*}0 \in U$ and thus $N(W \cup \{i{*}0\}) = N(W)$. If $\psi(i)\downarrow = 1$ then $V = W \cup \{i{*}0\}$ is a finite set which is incompatible to U. Thus $W_{N(V)} = V$ and $N(V) \neq N(W)$ since N is set driven; so $f(i) = 1$. The total recursive function f extends ψ in contradiction to the choice of ψ; thus such an IIM N cannot exist. ∎

Theorem 4.3 *Let $f \leq \mathrm{id}$ be a monotonic increasing function. Some family $\mathcal{L}$ is* SMON:*f learnable but not* LIM:*g learnable for any $g \not\geq f$.*

The set given after Theorem 6.1 is SMON$^{\mathrm{d}}$ learnable (and therefore also MON and MON$^{\mathrm{d}}$ learnable) but not learnable via a set driven IIM. So it remained the question, whether every SMON learnable set is learnable via a set driven IIM. Jain [8] refuted this conjecture and found a counterexample:

Theorem 4.4 [8] *Some* SMON *learnable class can not be inferred via a set driven* IIM.

5 Combining all Types of Restrictions

The next theorem shows that the hierarchy of the stronger monotonicity requirements is not changed by adding restrictions to the use of long term memory.

Theorem 5.1 *Let $\mathcal{A}$ and $\mathcal{B}$ be two learning criteria. Then every $\mathcal{A}$:f learnable $\mathcal{L}$ is also $\mathcal{B}$:f learnable iff there is an arrow (or a transitive chain of arrows) from $\mathcal{A}$ to $\mathcal{B}$ in the diagram below.*

$$\begin{tikzcd} \mathrm{SMON} \arrow[d] \arrow[rrd] & & \mathrm{SMON^d} \arrow[d] \arrow[lld] \\ \mathrm{MON} \arrow[rrd] & & \mathrm{MON^d} \arrow[d] \\ & & \mathrm{LIM} \end{tikzcd}$$

If $\mathcal{A} \not\to \mathcal{B}$, i.e., if there is neither a direct arrow nor a chain of arrows from $\mathcal{A}$ to $\mathcal{B}$, then

- *there is a class of sets which is $\mathcal{A}_k$:c learnable but not $\mathcal{B}$ learnable;*
- *there is a class of sets which is $\mathcal{A}_k$:c and $\mathcal{B}$:*id *learnable but not $\mathcal{B}$:f learnable for any $f \leq \mathrm{id} - 2$.*

It is always possible to take $k = 3$ mindchanges and $c = 2$ bits of long term memory.

The fact, that the classes witnessing the non-inclusions in Theorem 7.1 are learnable with a constant number of mindchanges, does not hold by a fluke. All classes learnable with constant long term memory are also learnable with constantly many mindchanges as the following Theorem 7.2 shows.

Theorem 5.2 *Let $\mathcal{A}$ be one of the inference criteria* CONV, SMON, SMON$^{\mathrm{d}}$, MON, MON$^{\mathrm{d}}$, LIM, WMON, WMON$^{\mathrm{d}}$*. If a class $\mathcal{L}$ is $\mathcal{A}$ learnable with constant long term memory, then $\mathcal{L}$ is also $\mathcal{A}$ learnable with a constant bound on both: memory and mindchanges. For any unbounded increasing function f there is a* SMON:*f learnable class which is not* LIM *learnable with a bounded number of mindchanges.*

Proof: Considering constant long term memory,in this proof it is more convinient to look upon the IIM as a finite state machine with input-alphabet $\Sigma^* \cup \{\#\}$. Since the alphabet is infinite, the IIM has a partial-recursive transition-function instead of a finite table. Let $M(\sigma)$ denote the guess of M after reading σ and let $m(\sigma)$ denote the state of M after reading σ; m and M are partial recursive and they are defined for all σ with $range(\sigma) \subseteq L$ for some $L \in \mathcal{L}$. This time, it is more suitable to measure the size of the long term memory in the number c of states of the finite automaton. It will turn out, that there is an IIM N inferring $\mathcal{L}$ with $2c-2$ mind changes.

The new IIM N simulates M and makes only a subsequence of M's guesses. If the given text for a language $L \in \mathcal{L}$ is $T = w_1 w_2 w_3 w_4 \ldots$ then N does not simulate the behaviour of M on T itself, but on a related text $T' = w_1\sigma_1 w_2\sigma_2 w_3\sigma_3\ldots$ with $range(\sigma_i) \subseteq \{w_1,\ldots,w_i\}$. N does not calculate the σ_i explicitly, N only uses the fact that such σ_i exist. The main idea is to ignore a new guess of M, if it is possible to force M to return to the last guess e of N via inserting such a σ_i. So N does not take all guesses of M and achieves the bound $2c-2$ on the number of mind changes.

Now the formal construction of N: N has two variables to store the current state and an older state of M; further for each state s of M, N has a counter b_s which takes the values $0,1,2$ and stores whether N has made 0, 1 or 2 guesses on transitions into s. So N's long term memory needs only to store one out of $c^2 \cdot 3^c$ possible values for the vector of these variables: constant long term memory is sufficient for N. N is initialized by $N(\lambda) = M(\lambda)$, $M(\lambda) = ?$ since M makes only guesses after reading a word of input. $d_0 = m(\lambda)$ is initialized to the initial state of M, $b_{m(\lambda),0} = 1$ and $b_{s,0} = 0$ for all other states s. Defining N inductively, assume that $N(w_1w_2\ldots w_n)$ is defined and input w_{n+1} is read. Further, let $e = N(w_1w_2\ldots w_j)$ be the last guess of N and d_j be the state of M after processing $w_1\sigma_1w_2\sigma_2\ldots w_j\sigma_j$.

Let $\sigma = w_1\sigma_1w_2\sigma_2\ldots w_n\sigma_n$. The construction has the invariant $d_n = m(\sigma)$. Let $q = m(\sigma w_{n+1})$ denote the state which M takes after reading w_{n+1} in state d_n.

- If $M(\sigma w_{n+1}) = ?$, i.e., if there is no new guess. Then N does also not make any new guess. So $N(w_1w_2\ldots w_nw_{n+1}) = ?$, $\sigma_{n+1} = \lambda$ and $d_{n+1} = q = m(\sigma w_{n+1}\lambda)$. All values b_s remain unchanged: $b_{s,n+1} = b_{s,n}$.

- If $M(\sigma w_{n+1}) = e'$ and $b_{q,n} < 2$. Then N makes the same guess, i.e., $\sigma_{n+1} = \lambda$ and $N(w_1w_2\ldots w_nw_{n+1}) = e'$. For bookkeeping, $b_{q,n+1} = b_{q,n} + 1$ while the other b_s remain unchanged (i.e., $b_{s,n+1} = b_{s,n}$). Again $d_{n+1} = q = m(\sigma w_{n+1}\lambda)$.

- If $M(\sigma w_{n+1}) = e'$ and $b_{q,n} = 2$. Then N makes no new guess, but N returns to the state d_j after the last guess: $N(w_1w_2\ldots w_nw_{n+1}) = ?$, $b_{s,n+1} = b_{s,n}$ and $d_{n+1} = d_j$. Note that there is a string σ_{n+1} such that $e = M(\sigma w_{n+1}\sigma_{n+1})$ and $d_{n+1} = d_j = m(\sigma w_{n+1}\sigma_{n+1})$.

It is easy to see that N needs only to know d_j, d_n, the values $b_{s,n}$ and the behaviour of M with input w_{n+1} in state d_n.

It remains to show that σ_{n+1} always exists in the third case. Let i denote the first stage such that $d_i = q$. If $q = m(\lambda)$, then $i = 0$ and $b_{q,i} = 1$. If $q \neq m(\lambda)$, then $b_{s,i} \leq 1$ since $b_{q,0}$ was initialized to 0 and is increased only via M going into stage q. Since all b_s are unchanged from stage j on, $b_{q,j} = 2$ and $i < j$. Now let $\sigma_{n+1} = w_{i+1}\sigma_{i+1}\ldots w_j\sigma_j$. So σ_{n+1} is a path from the state $q = d_i$ to the state d_j and $M(\sigma w_{n+1}\sigma_{n+1}) = M(w_1\sigma_1w_2\sigma_2\ldots w_j\sigma_j) = e$.

N makes at most $2c-1$ guesses, since N increases at each guess some b_s, $b_{m(\lambda)}$ is increased at most once and each other b_s is increase at most twice. Thus N makes at most $2c-2$ mind changes.

Let e be the last guess. M outputs e either infinitely often on T' or e is M's last guess on T'. Since T' is also a text for L, M has also to converge on T' to an index for L and thus, $L = W_e$. Since N makes a subsequence of M's guesses on the text T', N satisfies the same monotonicity criteria as M.

The family $\mathcal{L}$ to witness the second part consists of all finite sets $L_k = \{w_0, w_1, \ldots, w_k\}$ where $w_n = 0^{g(0)}1^{g(1)}\ldots n^{g(n)}$ for all n and g is some kind of inverse to f, i.e., $g(n) = \min\{m : f(m) \geq n\}$.

The SMON IIM needs only to store the maximal n such that a word w_n has been presented on the input. Whenever some w_m occurs in the input, the IIM checks whether $m \leq n$. If so, the IIM does nothing. If not, the IIM guesses $L_m = \{w_0, w_1, \ldots, w_m\}$ where L_m can easily be calculated from w_m. n takes the new value m.

$\mathcal{L}$ is not learnable with a bounded number of mind changes, since $L_0 \subset L_1 \subset L_2 \subset \ldots$ and the data of each L_k may be presented such that k mind changes are necessary to learn L_k. ∎

The IIM N from the first part of Theorem 5.2 uses more memory than M, but the amount of memory still is constant. The growth of the memory size from M to N is exponential: If M needs a long term memory of c states, then N needs $3^c \cdot c^2$ states. It might be, that the theorem is not optimal with respect to the increase of memory-usage.

But the result is optimal with respect to the number of mind changes, since the class $\mathcal{L} = \{\{0,1,\ldots,a\} : a = 2,\ldots,2c\}$ is on one hand learnable via an IIM whose long term memory consists of c states and on the other hand not learnable with less than $2c-2$ mind changes:

For each given IIM there is a text for $\{0,1,\ldots,2c\}$ such that the IIM outputs each guess $\{0,1,\ldots,a\}$ for $a = 2,3,\ldots,2c$. But $\mathcal{L}$ can be learned via an IIM using the c states $1,2,3,\ldots,c$. The state 1 is the initial state. Assume now that the IIM is in state i reads a number j. If $j < 2i$, the IIM makes no guess and stays in the state i. If $j \geq 2i$, the IIM guesses $\{0,1,2,\ldots,j\}$ and goes to stage $\lceil \frac{j}{2} \rceil$.

A further question is, whether there is a reversal of Theorem 5.2, i.e., whether bounded mind changes imply constant memory. But this fails: The family $\mathcal{L}$ containing the sets

$$\begin{array}{rcll} U_i & = & \{i*0, i*1, i*2, \ldots\} & \text{where } i \notin K; \\ V_i & = & \{i*0, i*1, i*2, \ldots, i*\varphi_i(i)\} & \text{where } i \in K. \end{array}$$

is SMON^d using only one mindchange. But $\mathcal{L}$ is neither WMON nor CONV learnable. Besides witnessing the non-inclusion SMON^d $\not\rightarrow$ WMON from Theorem 6.1, $\mathcal{L}$ witnesses that there are sets learnable via one mindchange which fail to be learnable under memory restrictions. Theorem 4.3 gives for every monotonic $f \leq \mathrm{id}$ an example of a family which on one hand is learnable via a SMON:f IIM making at most two mind changes and on the other hand is not LIM:g learnable for any $g \not\geq f$.

So it is suitable to consider a more restrictive precondition and a less restrictive hypothesis, namely classes learnable with 0 mind changes versus set driven inference. Note that learning with 0 mind changes implies that all monotonicity criteria hold.

Recall that an IIM which stores the last guess in its long term memory, i.e., which satisfies $S[\sigma] = \{M(\sigma)\}$ and $M(\sigma w) = \tilde{M}(M(\sigma), w)$, is called iterative. A modification of the proof of Theorem 5.2 shows that every class learnable with an IIM using a long term memory consisting of c states is also learnable via an IT IIM making at most $2c-2$ mind changes. The following theorem looks at the connections between learning with 0 mind changes, iterative learning and set driven learning.

Theorem 5.3 FIN *denotes the criterion to learn without any mind change.*
(a) *The class* $\mathcal{L} = \{W : |W| = 2\}$ *is* FIN *learnable but not* LIM:f *learnable for any* $f \not\geq \mathrm{id}$.
(b) *Every* FIN *learnable class is also* IT *learnable* [20].
(c) *Every* IT *learnable class* $\mathcal{H}$ *is learnable via a set driven* IIM.

Proof: The proof of part (b) can be found in Schäfer [20, p. 35].

Proof of (a): The algorithm, which outputs an index of $range(\sigma)$ iff $2 = |range(\sigma)|$ and makes no guess otherwise, obviously infers $\mathcal{L}$ without any mind change.

On the other hand consider a LIM:f IIM with $f(n) < n$ for some n. Then there are $c^{n+1}-1$ words of length up to n but the long term memory of the IIM can take only c^n different values after reading some word of length up to n. Thus there are two different words v and w such that both produce the same long term memory after being presented as first word of the input. It turns out, that the IIM will either fail to recognize $\{u, v\}$ or $\{u, w\}$ for some suitable u.

Proof of (c): Assume that the iterative IIM M learns the class $\mathcal{H}$. Since the content of the long term memory is identical to the last guess, one can assume that M never guesses ?. Furthermore note that $M(\sigma w) = M(\sigma)$ implies that $M(\sigma w^n) = M(\sigma)$ for all n.

Let T_W denote the ascending text with #'s of any given language W, i.e., if $W = \{w_1, w_2, \ldots\}$ is infinite then $T_W = w_1 \# w_2 \# \ldots$ and if $W = \{w_1, w_2, \ldots, w_n\}$ is finite then $T_W = \sigma_W \#^\infty$ where $\sigma_W = w_1 \# w_2 \# \ldots \# w_n$. Now the new set driven IIM N works as follows:

$$N(W) = \begin{cases} M(\sigma_W) & \text{if } M(\sigma_W) = M(\sigma_W \#); \\ \text{an index for } W & \text{otherwise.} \end{cases}$$

There are two cases:

Either $L \in \mathcal{H}$ is finite. Then it has to be shown that $N(L)$ must be an index of L: If $M(\sigma_L) \neq M(\sigma_L \#)$, this is true by the definition of N. Otherwise $M(\sigma_L) = M(\sigma_L \#^n)$ for all n and M converges on the text T_L to the the output $M(\sigma_L)$ of N. Since M infers L, $M(\sigma_L)$ is an index for L and also N infers L.

Or $L \in \mathcal{H}$ is infinite. Then it has to be shown that there is a finite set F such that $N(W) = e$ for all $W \in [F, L]$ and some index e for L. M converges on the text T_L to some index e of L. Let F be the first set such that $M(\eta) = e$ for all $\eta \in [\sigma_F, T_L]$. Then $\tilde{M}(e, w) = e$ for all $w \in L - F$ and $\tilde{M}(e, \#) = e$ since otherwise a further mind change would occur on the text T_L. If $W = F \cup \{w\}$ then $w \in L - F$. Now $N(W) = e$ since $\sigma_W = \sigma_F \# w$, $M(\sigma_W) = \tilde{M}(\tilde{M}(M(\sigma_F), \#), w) = \tilde{M}(\tilde{M}(e, \#), w) = \tilde{M}(e, w) = e$ and further $M(\sigma_W \#) = \tilde{M}(M(\sigma_W), \#) = \tilde{M}(e, \#) = e$. By induction $N(W) = e$ follows for all $W \in [F, L]$ and N infers L. ∎

6 Bounded Number of Mindchanges

This section deals with relations of the type "If $\mathcal{L}$ is $\mathcal{A}$ learnable then $\mathcal{L}$ is $\mathcal{B}$ learnable" between the monotonicity criteria in the general case without restrictions of long term memory. Thus also the criteria CONV, WMON and WMON$^{\mathrm{d}}$ are considered. Jain and Sharma [9] obtained these results for standard inference in the limit, here we also consider bounds on the number of mindchanges and give an overview by the following theorem:

Theorem 6.1 *Let* $\mathcal{A}$ *and* $\mathcal{B}$ *be two learning criteria. Then every* $\mathcal{A}$ *learnable* $\mathcal{L}$ *is also* $\mathcal{B}$ *learnable iff there is an arrow (or a transitive chain of arrows) from* $\mathcal{A}$ *to* $\mathcal{B}$ *in the diagram below.*

$$\begin{array}{ccccc} & \text{SMON} & & \text{SMON}^{\mathrm{d}} \\ \swarrow & \downarrow & \searrow\!\!\!\!\!\swarrow & \downarrow \\ \downarrow & \text{MON} & & \text{MON}^{\mathrm{d}} \\ \searrow & & \searrow & \downarrow \\ & \text{WMON} & \rightarrow & \text{WMON}^{\mathrm{d}} \\ & \uparrow\downarrow & & \uparrow\downarrow \\ & \text{CONV} & & \text{LIM} \end{array}$$

If in the diagram there is an arrow from $\mathcal{A}$ *to* $\mathcal{B}$ *or a transitive chain of arrows, then*

- *Any* $\mathcal{A}$ *learnable class of languages is* $\mathcal{B}$ *learnable;*
- *Any class of languages which is* $\mathcal{A}$ *learnable with* k *mindchanges is also* $\mathcal{B}$ *learnable with* k *mindchanges.*

Otherwise ($\mathcal{A} \not\rightarrow \mathcal{B}$)

- *there is a* $\mathcal{A}$ *learnable class of languages which is not* $\mathcal{B}$ *learnable;*
- *there is a class of languages which is* $\mathcal{A}$ *learnable with* 2 *mindchanges but which is not* $\mathcal{B}$ *learnable;*
- *there is a class of languages which is* $\mathcal{A}$ *learnable with* 2 *mindchanges and* $\mathcal{B}$ *learnable, but any* $\mathcal{B}$

learning algorithm makes an unbounded number of mindchanges.

All inclusions except WMON $\rightarrow$ CONV and LIM $\rightarrow$ WMON^d are obvious. While in many restricted contexts, MON learners are always also WMON learners [10, 11, 13, 15, 16, 17, 21], this natural relation fails in the general context.

7 Conclusion

First we considered learning r.e. languages from text under limitations on long term memory. It turned out that every superlinear bound can be tightened to a linear one without loosing inference power. Furthermore these classes of languages are learnable via a set driven inference machine. In the sublinear case there is a whole hierarchy. Second we found out that every class of languages is learnable under long term memory restriction is also conservatively learnable. So it was natrual to combine memory restrictions also with the other monotonicity requirements; the inclusions and non-inclusions on the criteria SMON, SMON^d, MON, MON^d and LIM are not changed by in addition requiring bounds on long term memory. If some IIM M infers $\mathcal{L}$ with constant amount of long term memory, then M can be translated into an IIM N which makes only a bounded number of mindchanges. Furthermore, if M satisfies some monotonicity requirement, so does N. On the other hand there is no reversal on this fact, i.e., if M makes at most one mindchange, it can not be translated into an equivalent N having a bound on long term memory. Third we showed that inclusion structure of monotonicity criteria does not change if in addition a bounded number of mindchanges is required.

Acknowledgments
The authors are thankful to John Case, Susanne Kaufmann, Martin Kummer and Mandayam Suraj for proofreading and helpful discussions.

References

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

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

[3] 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.

[4] Blum, M., and Blum, L. (1975), Towards a mathematical theory of inductive inference, *Information and Control*, **28**, pp. 125–155.

[5] Wexler, K., and Culicover, P.W. (1980), Formal principles of language acquisition. The MIT-Press, Cambridge Massachusets.

[6] Freivalds, R., Kinber, E., and Smith, C.H. (1993), On the impact of forgetting on learning machines, *in* "Proceedings of the 6th Annual ACM Conference on Computational Learning Theory", Santa Cruz, July 1993, pp. 165–174.

[7] Gold, E.M. (1967), Language identification in the limit, *Information and Control* **10**, pp. 447–474.

[8] Jain, S (1994) Private Communication.

[9] Jain, S., and Sharma, A. (1994), On monotonic strategies for learning r.e. languages, *in* "Proceedings of the 5th Workshop on Algorithmic Learning Theory", October 1994, pp. 349–364.

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

[11] Kapur, S. (1992), Monotonic language learning, *in* "Proceedings of the 3rd Workshop on Algorithmic Learning Theory", October 1992, Tokyo, JSAI, pp. 147–158.

[12] Kinber, E. (1994), Monotonicity versus Efficiency for Learning Languages from Texts, *in* "Proceedings of the 5th Workshop on Algorithmic Learning Theory", October 1994, pp. 395–406.

[13] Lange, S., and Zeugmann, T. (1992), Types of monotonic language learning and their characterization, *in* "Proceedings of the 5th Annual ACM Conference on Computational Learning Theory", Pittsburgh, July 1992, pp. 377–390, ACM Press, New York.

[14] Lange, S., and Zeugmann, T. (1993), Language Learning in Dependence on the Space of Hypotheses, *in* "Proceedings of the 6th Annual ACM Conference on Computational Learning Theory", Santa Cruz, July 1993, pp. 127–136, ACM Press, New York.

[15] Lange, S., and Zeugmann, T. (1993), Learning recursive languages with bounded mindchanges, *International Journal of Foundations of Computer Science* **4**, N02, 1993, pp. 157–178.

[16] Lange, S., and Zeugmann, T. (1994), A guided tour across the boundaries of learning recursive languages, unpublished manuscript.

[17] Lange, S., Zeugmann, T., and Kapur, S. (1992), Monotonic and dual monotonic language learning, to appear in *Theoretical Computer Science*. A prelimary version appeared as GOSLER-Report 14/94, TH Leipzig, FB Mathematik und Informatik, August 1992.

[18] Mukouchi, Y. (1992), Inductive inference with bounded mindchanges, *in* "Proceedings of the 3rd Workshop on Algorithmic Learning Theory", Tokyo, October 1992, JSAI, pp. 125–134.

[19] 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.

[20] Schäfer, G. (1984), Über Eingabeabhängigkeit und Komplexität von Inferenzstrategien. Thesis, Rheinisch-Westfälische Technische Hochschule Aachen, Mathematisch-Naturwissenschaftliche Fakultät.

[21] Zeugmann, T. (1993), Algorithmisches Lernen von Funktionen und Sprachen. Habitilationsschrift, Technische Hochschule Darmstadt, Fachbereich Informatik.