Learning with Temporary Memory

Steffen Lange[1], Samuel E. Moelius III[2], and Sandra Zilles[3]

[1] Fachbereich Informatik, Hochschule Darmstadt,
s.lange@fbi.h-da.de
[2] Department of Computer & Information Sciences, University of Delaware,
moelius@cis.udel.edu
[3] Department of Computing Science, University of Alberta,
zilles@cs.ualberta.ca

Abstract. In the inductive inference framework of learning in the limit, a variation of the bounded example memory (*Bem*) language learning model is considered. Intuitively, the new model constrains the learner's memory not only in *how much* data may be retained, but also in *how long* that data may be retained. More specifically, the model requires that, if a learner commits an example x to memory in some stage of the learning process, then there is some subsequent stage for which x *no longer* appears in the learner's memory. This model is called *temporary example memory* (*Tem*) learning. In some sense, it captures the idea that *memories fade*.

Many interesting results concerning the *Tem*-learning model are presented. For example, there exists a class of languages that can be identified by memorizing $k + 1$ examples in the *Tem* sense, but that can*not* be identified by memorizing k examples in the *Bem* sense. On the other hand, there exists a class of languages that can be identified by memorizing *just* 1 *example* in the *Bem* sense, but that can*not* be identified by memorizing *any number of examples* in the *Tem* sense. (The proof of this latter result involves an infinitary self-reference argument.) Results are also presented concerning the special cases of: learning *indexable* classes of languages, and learning (arbitrary) classes of *infinite* languages.

1 Introduction

The following is a common scenario in machine learning. A learner is repeatedly fed elements from an incoming stream of data. From this data, the learner must eventually generate a hypothesis that correctly identifies the contents of this stream of data. This is the case, for example, in many applications of neural networks (see [Mit97]).

In many cases, it would be impractical for a learning algorithm to *reconsider* all previously seen data when forming a new hypothesis. Thus, such learners are often designed to work in an *incremental* fashion, considering only the most recently presented datum, and possibly a few previously seen data that the learner considers to be significant.

This scenario has been studied formally by Lange and Zeugmann [LZ96] in the context of Gold-style language learning [Gol67]. Their model is called *bounded example memory* (*Bem*) learning. Intuitively, as the learner is fed elements from the incoming stream of data, the learner is allowed to commit up to k of these elements to memory, where k is *a priori* fixed. The learner may change which such elements are stored in its memory at any given time. However, any newly committed element *must* come from the incoming stream of data, *and*, the number of such elements can never exceed k. Among the results presented in [LZ96] is: for each k, there is a class of languages that can be identified by memorizing $k + 1$ examples, but that can*not* be identified by memorizing only k examples (Theorem 1 below). Further results on the *Bem*-learning model are obtained in [CJLZ99,CCJS07].

The *Bem*-learning model allows that any given example may be stored in the learner's memory *indefinitely*. However, most forms of computer memory are *volatile*, in that they require *energy* in order to retain their contents [RCN03]. Moreover, it has been observed in various areas of machine learning that the length of time for which data may be stored in a learner's memory can have an effect upon the capabilities of that learner (e.g., in reinforcement learning [LM92,McC96,Bak02] and in neural networks [HS97]).

Motivated by these observations, we consider a variation of the *Bem*-learning model in which the learner's memory is constrained not only in *how much* data may be stored, but also in *how long* that data may be stored. More specifically, we consider a model which requires that, if a learner commits an example x to memory in some stage of the learning process, then there is some subsequent stage for which x *no longer* appears in the learner's memory. We call this new model *temporary example memory* (*Tem*) learning. In some sense, this model captures the idea that *memories fade*.

Many interesting results concerning the *Tem*-learning model are presented. For example, there exists a class of languages that can be identified by memorizing $k + 1$ examples in the *Tem* sense, but that can*not* be identified by memorizing k examples in the *Bem* sense (Theorem 3). Thus, being able to store $k+1$ examples temporarily, can allow one to learn more than being able to store k example indefinitely. On the other hand, there exists a class of languages that can be identified by memorizing *just* 1 *example* in the *Bem* sense, but that can*not* be identified by memorizing *any number of examples* in the *Tem* sense (Theorem 4). Thus, being able to store just 1 example indefinitely, can allow one to learn more than being able to store any number of examples temporarily.

Results are also presented concerning the special cases of: learning *indexable* classes of languages, and learning (arbitrary) classes of *infinite* languages. For the case of indexable classes of languages, there exists such a class that can be identified by memorizing an arbitrary but finite number of examples in the *Bem* sense, but that can*not* be identified by memorizing an arbitrary but finite number of examples in the *Tem* sense (Theorem 5). In the case of classes of infinite languages, however, a completely different picture emerges. In particular, any such class that can be identified by memorizing an arbitrary but finite number of examples in the *Bem* sense, can also be identified by memorizing an arbitrary but finite number of examples in the *Tem* sense (Theorem 8). Intuitively, this latter result says that, when learning classes of infinite languages, restriction to temporary memory is, in fact, *not* a proper restriction.

In the context of both learning indexable classes of languages, and learning (arbitrary) classes of infinite languages, some problems remain open. These problems are stated formally in Sections 5 and 6.

Due to space constraints, many proofs are omitted or abbreviated. Complete proofs of all theorems can be found in [LMZ08].

2 Preliminaries

Computability-theoretic concepts not covered below are treated in [Rog67].

$\mathbb{N}$ denotes the set of natural numbers, $\{0, 1, 2, \ldots\}$. Lowercase italicized letters (e.g., a, b, c), with or without decorations, range over elements of $\mathbb{N}$, unless stated otherwise. In some cases, we treat $\mathbb{N}$ as the set of all strings over some finite alphabet Σ. In such cases, lowercase typewriter-font letters (e.g., $\mathtt{a}$, $\mathtt{b}$, $\mathtt{c}$) are used to denote alphabet symbols. For a symbol $\mathtt{a}$ and $n \in \mathbb{N}$, $\mathtt{a}^n$ denotes the string consisting of n repetitions of $\mathtt{a}$ (e.g., $\mathtt{a}^3 = \mathtt{aaa}$). For all strings x, $|x|$ denotes the length of x, i.e., the number of symbols in x.

A *language* is a subset of $\mathbb{N}$. Uppercase italicized letters (e.g., A, B, C), with or without decorations, range over languages. For all A, $\mathrm{Fin}(A)$ denotes the collection of all finite subsets of A. For all nonempty $A \subseteq \mathbb{N}$, $\min A$ denotes the minimum element of A, where $\min \emptyset \stackrel{\text{def}}{=} \infty$. For all nonempty, finite $A \subseteq \mathbb{N}$, $\max A$ denotes the maximum element of A, where $\max \emptyset \stackrel{\text{def}}{=} -1$. $\mathcal{L}$, with or without decorations, ranges over collections of languages.

Let $\#$ be a reserved symbol. For all languages L, *t is a text for L* $\stackrel{\text{def}}{\Leftrightarrow} t = (x_i)_{i \in \mathbb{N}}$, where $\{x_i \mid i \in \mathbb{N}\} \subseteq \mathbb{N} \cup \{\#\}$, and $L = \{x_i \mid i \in \mathbb{N}\} - \{\#\}$. For all L, $\mathit{Text}(L)$ denotes the set of all texts for L. For all texts $t = (x_i)_{i \in \mathbb{N}}$, $\mathit{content}(t) \stackrel{\text{def}}{=} \{x_i \mid i \in \mathbb{N}\} - \{\#\}$. For all texts t, and all $n \in \mathbb{N}$, $t[n]$ denotes the initial segment of t of length n.

For all one-argument partial functions ψ, and all $x \in \mathbb{N}$, $\psi(x)\downarrow$ denotes that $\psi(x)$ converges; $\psi(x)\uparrow$ denotes that $\psi(x)$ diverges. We use $\uparrow$ to denote the value of a divergent computation.

σ, with or without decorations, ranges over finite initial segments of texts for arbitrary languages. For all σ, $|\sigma|$ denotes the length of σ (equivalently, the size of the domain of σ). For all $\sigma = (x_i)_{i<n}$, $\mathit{content}(\sigma) \stackrel{\text{def}}{=} \{x_i \mid i < n\} - \{\#\}$. λ denotes the empty initial segment (equivalently, the everywhere divergent function). For all σ_0 and σ_1, $\sigma_0 \cdot \sigma_1$ denotes the concatenation of σ_0 and σ_1.

$\varphi_0, \varphi_1, \ldots$ denotes any fixed, acceptable numbering of all one-argument partial computable functions from $\mathbb{N}$ to $\mathbb{N}$. Φ denotes a fixed Blum complexity measure for φ. For each $i, s, x \in \mathbb{N}$,

$$\varphi_i^s(x) \stackrel{\text{def}}{=} \begin{cases} \varphi_i(x), & \text{if } [x < s \ \wedge \ \Phi_i(x) \leq s]; \\ \uparrow, & \text{otherwise.} \end{cases} \tag{1}$$

For each $i, s \in \mathbb{N}$, $W_i^s \stackrel{\text{def}}{=} \{x \mid \varphi_i^s(x)\downarrow\}$. For each $i \in \mathbb{N}$, $W_i \stackrel{\text{def}}{=} \bigcup_{s \in \mathbb{N}} W_i^s$. For each $s \in \mathbb{N}$, $W_\uparrow \stackrel{\text{def}}{=} W_\uparrow^s \stackrel{\text{def}}{=} \emptyset$.

An *inductive inference machine (IIM)* is a partial computable function whose inputs are initial segments of texts, and whose outputs are elements of $\mathbb{N}$ [OSW86]. $\mathbf{M}$, with or without decorations, ranges over IIMs.

Definitions 1 through 3 below introduce formally the Gold-style learning criteria of relevance to this paper. Therein, *Lim*, *Sdr*, and *It* are mnemonic for *limiting*, *set-driven*, and *iterative*, respectively. The first of these, *Lim*-learning (Definition 1 below), is the most fundamental. Intuitively, an IIM $\mathbf{M}$ is fed successively longer finite initial segments of a text for a target language L. $\mathbf{M}$ successfully identifies the language (from the given text) iff $\mathbf{M}$ converges to a hypothesis that correctly identifies the language (i.e., to a j such that $W_j = L$).

Definition 1 (Gold [Gol67]).

(a) Let $\mathbf{M}$ be an IIM, and let L be a language. $\mathbf{M}$ *LimTxt–identifies* L iff, for each text $t = (x_i)_{i \in \mathbb{N}} \in Text(L)$, there exists $n \in \mathbb{N}$ such that $W_{\mathbf{M}(t[n])} = L$ and $\mathbf{M}(t[i]) = \mathbf{M}(t[n])$ for all $i \geq n$.
(b) Let $\mathbf{M}$ be an IIM, and let $\mathcal{L}$ be a class of languages. $\mathbf{M}$ *LimTxt–identifies* $\mathcal{L}$ iff, for each $L \in \mathcal{L}$, $\mathbf{M}$ *LimTxt*–identifies L.
(c) $\mathit{LimTxt} = \{\mathcal{L} \mid (\exists \mathbf{M})[\mathbf{M}\ \mathit{LimTxt}\text{–identifies } \mathcal{L}]\}$.

The *Lim*-learning model allows that an IIM consider the entire initial segment of text presented to it when forming a new hypothesis. Thus, the IIM may consider: the *order* in which elements appear within that initial segment, and the *multiplicity* with which they appear. The set-driven (*Sdr*) learning model (Definition 2 below) restricts this. In particular, the *Sdr*-learning model requires that an IIM consider only the *contents* of any initial segment, and *not* the order or multiplicity of the elements therein.

Definition 2 (Wexler and Culicover [WC80]).

(a) Let $\mathbf{M}$ be an IIM, let L be a language, and let $M : \mathrm{Fin}(\mathbb{N}) \to \mathbb{N}$ be a partial computable function. $\mathbf{M}$ *SdrTxt–identifies* L *via* M iff (i) and (ii) below.
  (i) $\mathbf{M}$ *LimTxt*–identifies L.
  (ii) For each text $t = (x_i)_{i \in \mathbb{N}} \in Text(L)$, and each $i \in \mathbb{N}$, $M\big(content(t[i])\big) = \mathbf{M}(t[i])$.
(b) Let $\mathbf{M}$ be an IIM, and let $\mathcal{L}$ be a class of languages. $\mathbf{M}$ *SdrTxt–identifies* $\mathcal{L}$ iff there exists M such that, for each $L \in \mathcal{L}$, $\mathbf{M}$ *SdrTxt*–identifies L via M.
(c) $\mathit{SdrTxt} = \{\mathcal{L} \mid (\exists \mathbf{M})[\mathbf{M}\ \mathit{SdrTxt}\text{–identifies } \mathcal{L}]\}$.

Both of the preceding learning models allow that an IIM consider an *unbounded* number of elements when forming a new hypothesis. This does not seem practicable, in general, and motivates a desire for *memory limited* models of learning. Iterative (*It*) learning (Definition 3 below) is such a memory limited model. The *It*-model requires that an IIM consider *only* its most recently conjectured hypothesis, and the most recently occurring element of an initial segment of text. Thus, the IIM can*not*, in general, consider previously conjectured hypotheses, *nor* previously occurring elements of an initial segment of text.

Definition 3 (Wiehagen [Wie76]).

(a) Let $\mathbf{M}$ be an IIM, let L be a language, let $M : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be a partial computable function, and let $j_0 \in \mathbb{N}$. $\mathbf{M}$ *ItTxt–identifies* L *via* (M, j_0) iff (i) and (ii) below.
  (i) $\mathbf{M}$ *LimTxt*–identifies L.
  (ii) For each text $t = (x_i)_{i \in \mathbb{N}} \in Text(L)$, (α) through (γ) below.
    (α) For each $i \in \mathbb{N}$, $\mathbf{M}(t[i])\downarrow$.
    (β) $\mathbf{M}(t[0]) = j_0$.
    (γ) For each $i \in \mathbb{N}$, $\mathbf{M}(t[i+1]) = M\big(\mathbf{M}(t[i]), t(i)\big)$.

(b) Let $\mathbf{M}$ be an IIM, and let $\mathcal{L}$ be a class of languages. $\mathbf{M}$ *ItTxt–identifies* $\mathcal{L}$ iff there exists (M, j_0) such that, for each $L \in \mathcal{L}$, $\mathbf{M}$ *ItTxt*–identifies L via (M, j_0).

(c) $ItTxt = \{\mathcal{L} \mid (\exists \mathbf{M})[\mathbf{M}\ ItTxt\text{–identifies } \mathcal{L}]\}$.

Note that, in Definition 3(b), the behavior of $\mathbf{M}$ on any text t for a language in $\mathcal{L}$ is completely determined by j_0 and the behavior of M on j_0 and t. Thus, when referring to an iterative (or iterative-like) learner, we will, in some cases, refer only to (M, j_0) and avoid mention of $\mathbf{M}$ altogether. We do so similarly for set-driven learners (Definition 2). For iterative-like learning criteria that we define below (Definitions 4 and 5), we do so in terms of such (M, j_0) directly. In all such cases, it will be evident how to construct an appropriate IIM $\mathbf{M}$ from (M, j_0).

3 Bounded example memory ($\boldsymbol{Bem}$) learning

The following is a natural relaxation of It-learning called *k-bounded example-memory* (Bem_k) learning (Lange and Zeugmann [LZ96]). Recall that the It-learning model allows that an IIM consider the most recently occurring element of an initial segment of text, but *not* previously occurring elements. By contrast, the Bem_k-learning model allows that the IIM consider up to k such previously occurring elements, where $k \in \mathbb{N}^+$ is *a priori* fixed.

Definition 4 (Lange and Zeugmann [LZ96]). Let $k \in \mathbb{N}^+$ be fixed.

(a) Let $M : \big(\mathbb{N} \times \mathrm{Fin}(\mathbb{N})\big) \times \mathbb{N} \to \mathbb{N} \times \mathrm{Fin}(\mathbb{N})$ be a partial computable function, let $j_0 \in \mathbb{N}$, and let L be a language. (M, j_0) *Bem_kTxt–identifies* L iff, for each text $t = (x_i)_{i \in \mathbb{N}} \in Text(L)$, (i) through (iii) below.
  (i) For each $i \in \mathbb{N}$, $M_i(t)\downarrow$, where $M_0(t) = \langle j_0, \emptyset \rangle$ and $M_{i+1}(t) = M\big(M_i(t), x_i\big) = \langle j_{i+1}, X_{i+1} \rangle$.
  (ii) There exists $n \in \mathbb{N}$ such that $W_{j_n} = L$ and $j_i = j_n$ for all $i \geq n$.
  (iii) For each $i \in \mathbb{N}$, $X_{i+1} \subseteq X_i \cup \{x_i\}$ and $|X_{i+1}| \leq k$, where $X_0 = \emptyset$.

(b) Let (M, j_0) be as in (a), and let $\mathcal{L}$ be a class of languages. (M, j_0) *Bem_kTxt–identifies* $\mathcal{L}$ iff, for each $L \in \mathcal{L}$, (M, j_0) Bem_k*Txt*–identifies L.

(c) $Bem_kTxt = \{\mathcal{L} \mid (\exists M, j_0)[(M, j_0)\ Bem_kTxt\text{–identifies } \mathcal{L}]\}$.

For the remainder, let $\pi_1^2(\langle j, X\rangle) = j$ and $\pi_2^2(\langle j, X\rangle) = X$, for each $j \in \mathbb{N}$ and $X \in \mathrm{Fin}(\mathbb{N})$.

Note that Definition 4 allows an IIM to change the contents of its example memory infinitely often, even *after* it has converged to its final hypothesis. Thus, changing the contents of the example memory does *not* constitute a mind-change.

The classes $(\mathit{Bem}_k\mathit{Txt})_{k\in\mathbb{N}^+}$ defined in Definition 4(d) above form a proper hierarchy, as stated in the following theorem.

Theorem 1 (Lange and Zeugmann [LZ96]). For each $k \in \mathbb{N}^+$, $\mathit{Bem}_k\mathit{Txt} \subset \mathit{Bem}_{k+1}\mathit{Txt}$.

A natural variation of Lange and Zeugmann's model is to eliminate the restriction on the number of examples that can be memorized, i.e., to allow that the IIM store an arbitrary number of examples in its memory. We call the resulting learning model Bem_*-learning.

The formal definition of Bem_*-learning is obtained from Definition 4 by replacing Bem_k by Bem_* and by dropping the condition $|X_{i+1}| \leq k$ in (a)(iii).[4] This definition immediately implies the following.

Proposition 1. For each $k \in \mathbb{N}^+$, $\mathit{Bem}_k\mathit{Txt} \subseteq \mathit{Bem}_*\mathit{Txt}$.

Kinber and Stephan [KS95] studied a flexible notion of memory limited learning that subsumes our definition of Bem_*-learning. As an immediate consequence of their results, one obtains a characterization of Bem_*-learning in terms of set-driven learning (Definition 2 above). Recall that, with set-driven learning, the IIM can consider *neither* the order of the elements in the text, *nor* the multiplicity with which they appeared. However, the full set of previously seen examples is always accessible. The similarity to the definition of Bem_*-learning is obvious; nonetheless, the proof of the characterization is not completely straightforward. The reader is referred to [KS95] for details.

Theorem 2 (Kinber and Stephan [KS95]). $\mathit{SdrTxt} = \mathit{Bem}_*\mathit{Txt} \subset \mathit{LimTxt}$.

4 Temporary example memory (Tem) learning

This section introduces the temporary example memory (Tem) learning model. This model is a natural *restriction* of Bem-learning. It requires that, if a learner commits an example x to memory in some stage of the learning process, then there is some subsequent stage for which x *no longer* appears in the learner's memory.[5]

[4] N.B. The Bem_*-learning model does *not* afford the same capabilities to a learner as those provided by the *Lim*-learning model. Since the examples are stored in the learner's memory as a *set*, the learner can*not* consider the *order* in which those elements appeared, *nor* the *multiplicity* with which they appeared.

[5] As noted by one anonymous referee, one might reasonably allow elements occurring *infinitely often* in the text to remain in the learner's memory indefinitely. However, such a weakened restriction leads to a model *equivalent* to the one considered herein. The proof of this fact is omitted.

$$\begin{array}{ccccccc}
Bem_1Txt & \subset & Bem_2Txt & \subset & Bem_3Txt & \subset \cdots & Bem_*Txt \\
\cup & & \cup & & \cup & & \cup \\
Tem_1Txt & \subset & Tem_2Txt & \subset & Tem_3Txt & \subset \cdots & Tem_*Txt
\end{array}$$

$$Bem_1Txt \not\subseteq Tem_*Txt$$

Fig. 1. Summary of the results of Section 4.

Figure 1 summarizes the results of this section. The main results are the following. Theorem 3 says that there exists a class of languages that can be identified by memorizing $k+1$ examples in the Tem sense, but that can*not* be identified by memorizing k examples in the Bem sense. On the other hand, Theorem 4 says that there exists a class of languages that can be identified by memorizing *just* 1 *example* in the Bem sense, but that can*not* be identified by memorizing *any number of examples* in the Tem sense.

The following is the formal definition of Tem_k-learning. Note the addition of part (a)(iv), as compared to Definition 4.[6]

Definition 5. Let $k \in \mathbb{N}^+$ be fixed.

(a) Let $M : (\mathbb{N} \times \mathrm{Fin}(\mathbb{N})) \times \mathbb{N} \to \mathbb{N} \times \mathrm{Fin}(\mathbb{N})$ be a partial computable function, let $j_0 \in \mathbb{N}$, and let L be a language. (M, j_0) *Tem_kTxt–identifies* L iff, for each text $t = (x_i)_{i\in\mathbb{N}} \in Text(L)$, (i) through (iv) below.
   (i) For each $i \in \mathbb{N}$, $M_i(t)\downarrow$, where $M_0(t) = \langle j_0, \emptyset\rangle$ and $M_{i+1}(t) = M(M_i(t), x_i) = \langle j_{i+1}, X_{i+1}\rangle$.
   (ii) There exists $n \in \mathbb{N}$ such that $W_{j_n} = L$ and $j_i = j_n$ for all $i \geq n$.
   (iii) For each $i \in \mathbb{N}$, $X_{i+1} \subseteq X_i \cup \{x_i\}$ and $|X_{i+1}| \leq k$, where $X_0 = \emptyset$.
   (iv) For each $i \in \mathbb{N}$, there exists $i' \geq i$ such that $x_i \notin X_{i'+1}$.

(b) Let (M, j_0) be as in (a), and let $\mathcal{L}$ be a class of languages. (M, j_0) *Tem_kTxt–identifies* $\mathcal{L}$ iff, for each $L \in \mathcal{L}$, (M, j_0) Tem_kTxt–identifies L.

(c) $Tem_kTxt = \{\mathcal{L} \mid (\exists M, j_0)[(M, j_0)\ Tem_kTxt\text{–identifies } \mathcal{L}]\}$.

The preceding definition immediately implies the following.

Proposition 2. For each $k \in \mathbb{N}^+$, $Tem_kTxt \subseteq Bem_kTxt$.

The formal definition of Tem_*-learning is obtained from Definition 5 by replacing Tem_k by Tem_* and by dropping the condition $|X_{i+1}| \leq k$ in (a)(iii). Again, a few observations follow immediately.

Proposition 3. (a) For each $k \in \mathbb{N}^+$, $Tem_kTxt \subseteq Tem_*Txt$.
(b) $Tem_*Txt \subseteq Bem_*Txt$.

[6] For simplicity, Definition 5 allows that *when* an example is removed from memory be determined by the learner, as opposed to, say, by the environment. Technically, this gives the learner more control than absolutely necessary. However, this also makes the negative results obtained even more surprising (see, e.g., Theorem 4).

The following is the first main result of this section. Intuitively, it says that being able to store $k+1$ examples temporarily, can, in some cases, allow one to learn more than being able to store k examples indefinitely.

Theorem 3. *For each $k \in \mathbb{N}^+$, $\mathit{Tem}_{k+1}\mathit{Txt} - \mathit{Bem}_k\mathit{Txt} \neq \emptyset$.*

Proof (Sketch). Let $k \in \mathbb{N}^+$. For separating Tem_{k+1} and Bem_k we use a class that was already used in [LZ96] for the separation of Bem_{k+1} and Bem_k. We set $\Sigma = \{\mathtt{a}, \mathtt{b}\}$. For every $j, \ell_0, \dots, \ell_k \in \mathbb{N}$, let

$$L_{(j,\ell_0,\dots,\ell_k)} = \{\mathtt{a}^{j+1}\} \cup \{\mathtt{b}^z \mid z \leq j\} \cup \{\mathtt{b}^{\ell_0}, \dots, \mathtt{b}^{\ell_k}\}. \tag{2}$$

By $\mathcal{L}_k$ we denote the class containing $L = \{\mathtt{b}\}^*$ and all the languages $L_{(j,\ell_0,\dots,\ell_k)}$ for $j, \ell_0, \dots, \ell_k \in \mathbb{N}$. The proof that $\mathcal{L}_k \in \mathit{Tem}_{k+1}\mathit{Txt}$ is omitted, due to space constraints. That $\mathcal{L}_k \notin \mathit{Bem}_k\mathit{Txt}$ is proven in [LZ96]. □ (*Theorem 3*)

Theorem 3 has the following consequences.

Corollary 1. (a) *For each $k \in \mathbb{N}^+$, $\mathit{Tem}_k\mathit{Txt} \subset \mathit{Tem}_{k+1}\mathit{Txt}$.*
(b) *$\mathit{Tem}_*\mathit{Txt} - \bigcup_{k \in \mathbb{N}^+} \mathit{Bem}_k\mathit{Txt} \neq \emptyset$.*
(c) *$\bigcup_{k \in \mathbb{N}^+} \mathit{Bem}_k\mathit{Txt} \subset \mathit{Bem}_*\mathit{Txt}$.*
(d) *$\bigcup_{k \in \mathbb{N}^+} \mathit{Tem}_k\mathit{Txt} \subset \mathit{Tem}_*\mathit{Txt}$.*

In contrast to Theorem 3, restriction to temporary memory can have a significant effect upon a learner's capabilities, as demonstrated by our next main result. Intuitively, this result says that being able to store just 1 example indefinitely, can allow one to learn more than being able to store any number of examples temporarily. The proof involves an infinitary self-reference argument.

Theorem 4. *$\mathit{Bem}_1\mathit{Txt} - \mathit{Tem}_*\mathit{Txt} \neq \emptyset$.*

Proof (Sketch). Let $\mathcal{L} =$

$$\begin{aligned} &\{L \mid 0 \notin L \ \wedge \ (\forall e \in L)[W_e = L]\} \\ \cup &\big\{L \mid 0 \in L \ \wedge \ (\exists u \in L - \{0\})\big[\ \ W_u = \emptyset \\ &\qquad\qquad\qquad\qquad\qquad \wedge (\forall e \in L - \{0,u\})[W_e = L - \{0,u\}]\big]\big\}. \end{aligned} \tag{3}$$

The proof that $\mathcal{L} \in \mathit{Bem}_1\mathit{Txt}$ is omitted, due to space constraints. The proof that $\mathcal{L} \notin \mathit{Tem}_*\mathit{Txt}$ follows. By way of contradiction, let $\mathbf{M}$ be such that $\mathbf{M}$ $\mathit{Tem}_*\mathit{Txt}$-identifies $\mathcal{L}$. By the Operator Recursion Theorem [Cas74,Cas94], there exist *distinct* φ-programs $(e_j)_{j \in \mathbb{N}}$, *none* of which are 0, and whose behavior is determined by the construction in Figure 2. In conjunction with $(e_j)_{j \in \mathbb{N}}$, a series of finite sequences $(\sigma^s)_{s \in \mathbb{N}}$ is constructed. Note that, in the construction of $(\sigma^s)_{s \in \mathbb{N}}$, σ^{s+1} is defined $\Leftrightarrow$ stage s is exited. So, if there is a *least* s_0 such that stage s_0 is *not* exited, then, for all $s' \geq s_0$, let $\sigma^{s'+1} = \sigma^{s_0}$.

Claim 1. (a) through (d) below.

(a) $(\forall s \in \mathbb{N})[\sigma^s \subseteq \sigma^{s+1}]$.
(b) $(\forall s \in \mathbb{N})\big[[s = 0 \vee \text{stage } s-1 \text{ is exited}] \Rightarrow (\forall j < 2s)[\mathit{content}(\sigma^s) \subseteq W_{e_j}]\big]$.
(c) $(\forall j \in \mathbb{N})[W_{e_j} \subseteq \bigcup_{s \in \mathbb{N}} \mathit{content}(\sigma^s)]$.
(d) $(\forall s \in \mathbb{N})[\mathit{content}(\sigma^s) \subseteq \{e_j \mid j < 2s\}]$.

Proof of Claim. Clear by the construction of $(e_j)_{j \in \mathbb{N}}$ and $(\sigma^s)_{s \in \mathbb{N}}$. □ (*Claim 1*)

Set $\sigma^0 = \lambda$, and execute stages $s = 0, 1, ...$, successively, as follows.

STAGE s. Find the *least* $m \in \mathbb{N}$ (if any) for which one of the following conditions applies, and act accordingly.

COND. (i) $(\exists i \in \{0,1\})[\mathbf{M}(\sigma^s \cdot e_{2s+i} \cdot \#^m)\uparrow]$. Go into an infinite loop.

COND. (ii) $\big[\neg\text{(i)} \ \wedge \ (\exists i \in \{0,1\})[(\pi_1^2 \circ \mathbf{M})(\sigma^s \cdot e_{2s+i} \cdot \#^m) \neq (\pi_1^2 \circ \mathbf{M})(\sigma^s)]\big]$.
(a) For the *least* $i \in \{0,1\}$ satisfying the condition, set $\sigma^{s+1} = \sigma^s \cdot e_{2s+i} \cdot \#^m$.
(b) For each $j < 2s+2$, list $content(\sigma^{s+1})$ into W_{e_j}.
(c) Proceed to stage $s+1$.

COND. (iii) $\big[\neg\text{(i)-(ii)} \ \wedge \ (\forall i \in \{0,1\})[(\pi_2^2 \circ \mathbf{M})(\sigma^s \cdot e_{2s+i} \cdot \#^m) = \emptyset]\big]$.
(a) Set $\sigma^{s+1} = \sigma^s$.
(b) Terminate the construction.

COND. (iv) $\big[\neg\text{(i)-(iii)} \ \wedge \ m > 0 \ \wedge \ (\exists i \in \{0,1\})[(\pi_2^2 \circ \mathbf{M})(\sigma^s \cdot e_{2s+i} \cdot \#^m) = (\pi_2^2 \circ \mathbf{M})(\sigma^s \cdot e_{2s+i} \cdot \#^{m-1}) \neq \emptyset]\big]$.
(a) For the *least* $i \in \{0,1\}$ satisfying the condition, set $\sigma^{s+1} = \sigma^s \cdot e_{2s+i} \cdot \#^m$.
(b) For each $j < 2s+2$, list $content(\sigma^{s+1})$ into W_{e_j}.
(c) Terminate the construction.

Fig. 2. The construction of $(e_j)_{j \in \mathbb{N}}$ and $(\sigma^s)_{s \in \mathbb{N}}$ in the proof that $\mathcal{L} \notin \mathit{Tem}_*\mathit{Txt}$ (part of Theorem 4). Note: nothing is listed into W_{e_j}, for any j, aside from the above.

Consider the following cases.

CASE (I) $(\exists s \in \mathbb{N})(\forall m \in \mathbb{N})$[*none* of COND. (i)-(iv) apply for m in stage s]. Then, for all $m \in \mathbb{N}$, (**i**) through (**iv**) below.

(**i**) $(\forall i \in \{0,1\})[\mathbf{M}(\sigma^s \cdot e_{2s+i} \cdot \#^m)\downarrow]$.
(**ii**) $(\forall i \in \{0,1\})[(\pi_1^2 \circ \mathbf{M})(\sigma^s \cdot e_{2s+i} \cdot \#^m) = (\pi_1^2 \circ \mathbf{M})(\sigma^s)]$.
(**iii**) $(\exists i \in \{0,1\})[(\pi_2^2 \circ \mathbf{M})(\sigma^s \cdot e_{2s+i} \cdot \#^m) \neq \emptyset]$.
(**iv**) $m > 0 \ \Rightarrow \ (\forall i \in \{0,1\})[\ (\pi_2^2 \circ \mathbf{M})(\sigma^s \cdot e_{2s+i} \cdot \#^m) \neq (\pi_2^2 \circ \mathbf{M})(\sigma^s \cdot e_{2s+i} \cdot \#^{m-1}) \vee (\pi_2^2 \circ \mathbf{M})(\sigma^s \cdot e_{2s+i} \cdot \#^{m-1}) = \emptyset]$.

By (**i**) and (**ii**), clearly, for all $i \in \{0,1\}$,

$$(\pi_2^2 \circ \mathbf{M})(\sigma^s \cdot e_{2s+i}) \supseteq (\pi_2^2 \circ \mathbf{M})(\sigma^s \cdot e_{2s+i} \cdot \#) \supseteq (\pi_2^2 \circ \mathbf{M})(\sigma^s \cdot e_{2s+i} \cdot \#^2) \supseteq \cdots . \quad (4)$$

Since, for all σ, $(\pi_2^2 \circ \mathbf{M})(\sigma)$ is a *finite* set, both of the sequences corresponding to (4) must eventually reach a fixpoint. But, clearly, by (**iii**) and (**iv**), at least one such sequence does *not* reach a fixpoint (a contradiction).

CASE (II) $(\exists s, m \in \mathbb{N})$[COND. (i) applies for m in stage s]. Then, clearly,

$$(\forall s')[\text{stage } s' \text{ is exited} \ \Leftrightarrow \ s' < s]. \quad (5)$$

Thus, for all $j < 2s$,

$$\begin{aligned} content(\sigma^s) &\subseteq W_{e_j} && \{\text{by (5) and Claim 1(b)}\} \\ &\subseteq content(\sigma^s) && \{\text{by (a) and (c) of Claim 1, and (5)}\} \\ &\subseteq \{e_j \mid j < 2s\} && \{\text{by Claim 1(d)}\}. \end{aligned} \quad (6)$$

Clearly, by the construction of $(e_j)_{j\in\mathbb{N}}$,

$$(\forall i \in \{0,1\})[W_{e_{2s+i}} = \emptyset]. \tag{7}$$

Let $i \in \{0,1\}$ be *least* such that

$$\mathbf{M}(\sigma^s \cdot e_{2s+i} \cdot \#^m)\uparrow. \tag{8}$$

Let t' be such that $t' = \sigma^s \cdot e_{2s+i} \cdot \#^m \cdot 0 \cdot \# \cdot \# \cdot \cdots$. Let $L' = \mathit{content}(t')$. By (6) and (7), clearly, L' is a language in $\mathcal{L}$ of the second type in (3) (where, $u = e_{2s+i}$). But, by (8), $\mathbf{M}$ does *not* $\mathit{Tem}_*\mathit{Txt}$-identify L' from t' (a contradiction).

Case (III) $(\exists s, m \in \mathbb{N})$[Cond. (iii) applies for m in stage s]. Then, clearly,

$$(\forall s')[\text{stage } s' \text{ is exited } \Leftrightarrow\ s' \leq s]. \tag{9}$$

Thus, for all $j < 2s$,

$$\begin{aligned}
\mathit{content}(\sigma^s) &= W_{e_j} && \{\text{by (9) and Claim 1(b)}\}\\
&\subseteq \mathit{content}(\sigma^{s+1}) && \{\text{by (a) and (c) of Claim 1, and (9)}\}\\
&= \mathit{content}(\sigma^s) && \{\text{by the case and the constr. of } (\sigma^s)_{s\in\mathbb{N}}\}\\
&\subseteq \{e_j \mid j < 2s\} && \{\text{by Claim 1(d)}\}.
\end{aligned} \tag{10}$$

Clearly, by the construction of $(e_j)_{j\in\mathbb{N}}$,

$$(\forall i \in \{0,1\})[W_{e_{2s+i}} = \emptyset]. \tag{11}$$

Note that part of Cond. (iii) is that Cond. (ii) does *not* apply. Thus,

$$(\forall i \in \{0,1\})[\mathbf{M}(\sigma^s \cdot e_{2s+i} \cdot \#^m) = \langle(\pi_1^2 \circ \mathbf{M})(\sigma^s), \emptyset\rangle]. \tag{12}$$

For all $i \in \{0,1\}$, let t'_i be such that $t'_i = \sigma^s \cdot e_{2s+i} \cdot \#^m \cdot 0 \cdot \# \cdot \# \cdot \cdots$. For all $i \in \{0,1\}$, let $L'_i = \mathit{content}(t'_i)$. By (10) and (11), clearly, L'_0 and L'_1 are *distinct* languages in $\mathcal{L}$ of the second type in (3) (where, $u = e_{2s}$ for L'_0, and $u = e_{2s+1}$ for L'_1). But, by (12), $\mathbf{M}$ can*not* distinguish L'_0 and L'_1 (a contradiction).

Case (IV) $(\exists s, m \in \mathbb{N})$[Cond. (iv) applies for m in stage s]. Let $i \in \{0,1\}$ be *least* such that

$$(\pi_2^2 \circ \mathbf{M})(\sigma^s \cdot e_{2s+i} \cdot \#^m) = (\pi_2^2 \circ \mathbf{M})(\sigma^s \cdot e_{2s+i} \cdot \#^{m-1}) \neq \emptyset. \tag{13}$$

Note that, part of Cond. (iv) is that Cond. (ii) does *not* apply. Furthermore, by the case, $m > 0$. Thus, it must also be that Cond. (ii) does *not* apply for $m-1$ (in stage s). Consequently,

$$\begin{aligned}
&\mathbf{M}(\sigma^s \cdot e_{2s+i} \cdot \#^m)\\
&= \langle(\pi_1^2 \circ \mathbf{M})(\sigma^s \cdot e_{2s+i} \cdot \#^m), (\pi_2^2 \circ \mathbf{M})(\sigma^s \cdot e_{2s+i} \cdot \#^m)\rangle && \{\text{immediate}\}\\
&= \langle(\pi_1^2 \circ \mathbf{M})(\sigma^s \cdot e_{2s+i} \cdot \#^m), (\pi_2^2 \circ \mathbf{M})(\sigma^s \cdot e_{2s+i} \cdot \#^{m-1})\rangle && \{\text{by (13)}\}\\
&= \langle(\pi_1^2 \circ \mathbf{M})(\sigma^s), (\pi_2^2 \circ \mathbf{M})(\sigma^s \cdot e_{2s+i} \cdot \#^{m-1})\rangle && \{\text{by } \neg\text{(ii) for } m\}\\
&= \langle(\pi_1^2 \circ \mathbf{M})(\sigma^s \cdot e_{2s+i} \cdot \#^{m-1}), (\pi_2^2 \circ \mathbf{M})(\sigma^s \cdot e_{2s+i} \cdot \#^{m-1})\rangle && \{\text{by } \neg\text{(ii)}\\
&&& \quad\text{for } m-1\}\\
&= \mathbf{M}(\sigma^s \cdot e_{2s+i} \cdot \#^{m-1}) && \{\text{immediate}\}.
\end{aligned}$$

Clearly, then, for all $n \geq m$,

$$\mathbf{M}(\sigma^s \cdot e_{2s+i} \cdot \#^n) = \mathbf{M}(\sigma^s \cdot e_{2s+i} \cdot \#^{m-1}). \tag{14}$$

Next, note that, by the construction of $(\sigma^s)_{s\in\mathbb{N}}$,

$$\sigma^{s+1} = \sigma^s \cdot e_{2s+i} \cdot \#^m. \tag{15}$$

Clearly,

$$(\forall s')[\text{stage } s' \text{ is exited} \;\Leftrightarrow\; s' \leq s]. \tag{16}$$

Thus, for all $j < 2s+2$,

$$\begin{aligned} \mathit{content}(\sigma^{s+1}) &\subseteq W_{e_j} && \{\text{by (16) and Claim 1(b)}\} \\ &\subseteq \mathit{content}(\sigma^{s+1}) && \{\text{by (a) and (c) of Claim 1, and (16)}\} \\ &\subseteq \{e_j \mid j < 2s+2\} && \{\text{by Claim 1(d)}\}. \end{aligned} \tag{17}$$

Let t be such that $t = \sigma^{s+1} \cdot \# \cdot \# \cdot \cdots$. Let $L = \mathit{content}(t)$. By (17), clearly, L is a language in $\mathcal{L}$ of the first type in (3). But, by (13), (14), and (15), $\mathbf{M}$ does *not* $\mathit{Tem}_*\mathit{Txt}$-identify L from t (a contradiction).

Case (V) [¬(I)-(IV)]. Then, clearly,

$$(\forall s \in \mathbb{N})(\exists m \in \mathbb{N})[\text{Cond. (ii) applies for } m \text{ in stage } s]. \tag{18}$$

Let $t = \lim_{s\to\infty} \sigma^s$. By Claim 1(a), t is well-defined, and, by (18) and the construction of $(\sigma^s)_{s\in\mathbb{N}}$, t is total. Clearly,

$$(\forall s \in \mathbb{N})[\text{stage } s \text{ is exited}]. \tag{19}$$

Thus, for all $j \in \mathbb{N}$,

$$\begin{aligned} \mathit{content}(t) &= \textstyle\bigcup_{s\in\mathbb{N}} \mathit{content}(\sigma^s) && \{\text{immediate}\} \\ &\subseteq W_{e_j} && \{\text{by (19), and (a) and (b) of Claim 1}\} \\ &\subseteq \textstyle\bigcup_{s\in\mathbb{N}} \mathit{content}(\sigma^s) && \{\text{by Claim 1(c)}\} \\ &\subseteq \{e_j \mid j \in \mathbb{N}\} && \{\text{by Claim 1(d)}\}. \end{aligned} \tag{20}$$

By (20), $\mathit{content}(t)$ is a language in $\mathcal{L}$ of the first type in (3). But, by (18), $\mathbf{M}$ *never* reaches a final conjecture on t (a contradiction). □ (*Theorem 4*)

The preceding result, along with Theorem 1 and Propositions 2 and 3, yields the following corollary.

Corollary 2. (a) For each $k \in \mathbb{N}^+$, $\mathit{Tem}_k\mathit{Txt} \subset \mathit{Bem}_k\mathit{Txt}$.
(b) $\mathit{Tem}_*\mathit{Txt} \subset \mathit{Bem}_*\mathit{Txt}$.

5 *Tem*-learning of *indexable* classes of languages

In this section, we consider the special case of *Tem*-learning of indexable classes of languages. A class of languages $\mathcal{L}$ is indexable iff (by definition) there exists a computable function $d : \mathbb{N} \times \mathbb{N} \to \{0,1\}$ such that $\mathcal{L} = \{L_i \mid i \in \mathbb{N}\}$ where, for each $i \in \mathbb{N}$, $L_i = \{x \in \mathbb{N} \mid d(i,x) = 1\}$ [LZZ08]. Many interesting *and natural* classes of languages are indexable. For example, the classes of regular and context free languages [HMU01] are each indexable.

The next two results say that two of the important separation results obtained in Section 4 are witnessed by indexable classes of languages.

Corollary 3 (of the proof of Theorem 3). For each $k \in \mathbb{N}^+$, there is an indexable class of languages $\mathcal{L}_k$ such that $\mathcal{L}_k \in \mathit{Tem}_{k+1}\mathit{Txt} - \mathit{Bem}_k\mathit{Txt}$.

Proof of Corollary. One need only observe that each of the $\mathcal{L}_k$ constructed in the proof of Theorem 3 is an indexable class. □ (*Corollary 3*)

Theorem 5. There is an indexable class of languages $\mathcal{L} \in \mathit{Bem}_*\mathit{Txt} - \mathit{Tem}_*\mathit{Txt}$.

Proof (Sketch). Let $\langle\cdot,\cdot\rangle : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be any 1-1, onto, computable function. For all $A, B \subseteq \mathbb{N}$, let $A \times B = \{\langle a, b\rangle \mid a \in A \ \wedge\ b \in B\}$. Let $\mathcal{L}$ be such that

$$\mathcal{L} = \big\{\{e\} \times A \mid e \in \mathbb{N} \ \wedge\ 0 \in A \ \wedge\ A \in \mathrm{Fin}(\mathbb{N})\big\} \cup \{L_e \mid e \in \mathbb{N}\}, \tag{21}$$

where, for each $e \in \mathbb{N}$, $L_e = \bigcup_{s\in\mathbb{N}} \mathit{content}(\sigma_e^s)$, and $(\sigma_e^s)_{s\in\mathbb{N}}$ is constructed as follows.

Set $\sigma_e^0 = \langle e, 1\rangle$, and execute stages $s = 0, 1, ...,$ successively, as follows.

Stage s. Act according to the following (decidable) conditions, then go to stage $s + 1$.

Cond. (i) $(\exists n \le s)[\Phi_e(\sigma_e^s \cdot \#^n) \le s \ \wedge\ (\pi_2^2 \circ \varphi_e)(\sigma_e^s \cdot \#^n) = \emptyset]$. For the *least* $n \in \mathbb{N}$ satisfying the condition, set $\sigma_e^{s+1} = \sigma_e^s \cdot \#^n \cdot \langle e, s + 2\rangle$.

Cond. (ii) $[\neg(\mathrm{i})]$. Set $\sigma_e^{s+1} = \sigma_e^s$.

Note that, for all $e, s \in \mathbb{N}$, $\langle e, s + 2\rangle \in L_e \Leftrightarrow$ Cond. (i) applies in stage s in the construction of $(\sigma_e^s)_{s\in\mathbb{N}}$. Furthermore, it is clearly the case that $\{\langle e, 1\rangle\} \subseteq L_e \subseteq \{e\} \times (\mathbb{N} + 1)$. Thus, each L_e is computable. By only slightly more reasoning, it can be seen that $\mathcal{L}$ is an indexable class.

It is easily seen that $\mathcal{L} \in \mathit{SdrTxt}$. Thus, by Theorem 2, $\mathcal{L} \in \mathit{Bem}_*\mathit{Txt}$. The proof that $\mathcal{L} \notin \mathit{Tem}_*\mathit{Txt}$ is omitted, due to space constraints. □ (*Theorem 5*)

It is currently open whether or not the remaining separation results of Section 4 can be witnessed by indexable classes of languages.

Problem 1. Let $k \in \mathbb{N}^+$, $\mathfrak{A} \in \{\mathit{Bem}_1\mathit{Txt}, ..., \mathit{Bem}_k\mathit{Txt}\}$, and $\mathfrak{B} \in \{\mathit{Tem}_k\mathit{Txt}, \mathit{Tem}_{k+1}\mathit{Txt}, ..., \mathit{Tem}_*\mathit{Txt}\}$. Is there an indexable class of languages $\mathcal{L} \in \mathfrak{A} - \mathfrak{B}$?

6 *Tem*-learning of classes of *infinite* languages

In this section, we consider the special case of *Tem*-learning of classes of infinite languages. Our main result of this section, Theorem 8, says that any class of infinite languages that can be identified by memorizing an arbitrary but finite number of examples in the *Bem* sense, can also be identified by memorizing an arbitrary but finite number of examples in the *Tem* sense.

Our first result of this section says that one of the important separation results obtained in Section 4 is witnessed by a class of infinite languages.

Theorem 6. For each $k \in \mathbb{N}^+$, there exists a class $\mathcal{L}_k$ of infinite languages such that $\mathcal{L}_k \in \mathit{Tem}_{k+1}\mathit{Txt} - \mathit{Bem}_k\mathit{Txt}$.

Proof (Sketch). Let $k \in \mathbb{N}^+$. Fix $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$. The witnessing class can be defined by taking the class $\mathcal{L}_k$ used in the proof of Theorem 3 and by adding the infinite set $\{\mathtt{c}\}^*$ to every language in this class. Further details are omitted. □ (*Theorem 6*)

Before presenting our next main result, it is worth recalling the following.

Theorem 7 (Osherson, Stob, and Weinstein [OSW86]). Let $\mathcal{L}$ be any class of infinite languages. Then, $\mathcal{L} \in \mathit{LimTxt}$ iff $\mathcal{L} \in \mathit{SdrTxt}$.

Note that Theorems 2 and 7 have the following corollary.

Corollary 4 (of Theorems 2 and 7). Let $\mathcal{L}$ be any class of infinite languages. Then, $\mathcal{L} \in \mathit{LimTxt}$ iff $\mathcal{L} \in \mathit{Bem}_*\mathit{Txt}$.

Thus, Bem_*-learning is *not* a proper restriction when learning classes of infinite languages. This is in contrast to Theorem 2 which also says that Bem_*-learning *is* a proper restriction when learning classes of arbitrary languages.

Our next main result says that Tem_*-learning is equivalent to Bem_*-learning when learning classes of infinite languages. Thus, by Corollary 4, Tem_*-learning is similarly *not* a proper restriction when learning classes of infinite languages.

Theorem 8. Let $\mathcal{L}$ be any class of infinite languages. Then, $\mathcal{L} \in \mathit{Bem}_*\mathit{Txt}$ iff $\mathcal{L} \in \mathit{Tem}_*\mathit{Txt}$.

Proof (Sketch). By Proposition 3, it suffices to show that, for each class of infinite languages $\mathcal{L}$, if $\mathcal{L} \in \mathit{Bem}_*\mathit{Txt}$, then $\mathcal{L} \in \mathit{Tem}_*\mathit{Txt}$. So, let $\mathcal{L}$ be a class of infinite languages, and suppose that $\mathcal{L} \in \mathit{Bem}_*\mathit{Txt}$. An M' is constructed such that M' $\mathit{Tem}_*\mathit{Txt}$-identifies $\mathcal{L}$. Due to space constraints, we give only the construction of M', and not the proof of its correctness.

By Theorem 2, there exists M such that M SdrTxt-identifies $\mathcal{L}$. Without loss of generality, suppose that $M(\emptyset)\downarrow$. Let p_M be such that, for all finite $A \subseteq \mathbb{N}$, $\varphi_{p_M}(A) = M(A)$. By 1-1 s-m-n [Rog67], there exists a 1-1 computable function f such that, for all finite $A, B \subseteq \mathbb{N}$, and all $k \in \{0, 1\}$, $W_{f(A,B,k)} = W_{M(A)}$.

For all $L \in \mathcal{L}$, all $t = (x_i)_{i \in \mathbb{N}} \in \mathit{Text}(L)$, and all $i \in \mathbb{N}$, let M' be as follows. $M'_0(t) = \langle f(\emptyset, \emptyset, 0), \emptyset \rangle$ and $M'_{i+1}(t) = M'\big(\langle f(A_i, B_i, k_i), X_i \rangle, x_i\big) = \langle f(A_{i+1}, B_{i+1}, k_{i+1}), X_{i+1} \rangle$, where A_{i+1}, B_{i+1}, k_{i+1}, and X_{i+1} are determined as in Figure 3.

The remainder of the proof is omitted. □ (*Theorem 8*)

Corollary 5. Let $\mathcal{L}$ be any class of infinite languages. Then, $\mathcal{L} \in \mathit{LimTxt}$ iff $\mathcal{L} \in \mathit{Tem}_*\mathit{Txt}$.

It is currently open whether or not the remaining separation results of Section 4 can be witnessed by classes of infinite languages.

Problem 2. Let $k \in \mathbb{N}^+$, $\mathfrak{A} \in \{\mathit{Bem}_1\mathit{Txt}, ..., \mathit{Bem}_k\mathit{Txt}\}$, and $\mathfrak{B} \in \{\mathit{Tem}_k\mathit{Txt}, \mathit{Tem}_{k+1}\mathit{Txt}, ..., \mathit{Tem}_*\mathit{Txt}\}$. Is there a class of infinite languages $\mathcal{L} \in \mathfrak{A} - \mathfrak{B}$?

$A_0 = B_0 = X_0 = \emptyset$ and $k_0 = 0$. For each $i \in \mathbb{N}$, $A_{i+1} = A_i$, $B_{i+1} = B_i$, $k_{i+1} = k_i$, and $X_{i+1} = X_i$, unless stated otherwise.

if $x_i \neq \#$ **then**
  let $B_i^* = \begin{cases} B_i, & \text{if } k_i = 0; \\ W_{M(A_i)}^{s_i^{\min}}, & \text{if } k_i = 1, \text{ where } s_i^{\min} = \min\{s \mid (W_{M(A_i)}^{s+1} \cap X_i) \neq \emptyset\}; \end{cases}$
  /* For the latter case, it can be shown that $s_i^{\min} < \infty$. */
  let $X_i^+ = (X_i \cup \{x_i\})$;
  let $C_i = (B_i^* \cup X_i^+)$;
  let $S_i = \big\{s \leq \max(C_i) \mid B_i^* \subseteq W_{M(A_i)}^s \subseteq C_i \;\wedge\; \big(W_{M(A_i)}^{s+1} \cap (X_i^+ - W_{M(A_i)}^s)\big) \neq \emptyset\big\}$;
  if $(\exists A')\big[B_i \subseteq A' \subseteq C_i \;\wedge\; \varphi_{p_M}^{\max(C_i)}(A')\!\downarrow \neq \varphi_{p_M}^{\max(C_i)}(A_i)\!\downarrow\big]$ **then**
    $A_{i+1} \leftarrow A';\; B_{i+1} \leftarrow C_i;\; k_{i+1} \leftarrow 0;\; X_{i+1} \leftarrow \emptyset$;
  else if $S_i \neq \emptyset$ **then**
    $k_{i+1} \leftarrow 1;\; X_{i+1} \leftarrow (X_i^+ - W_{M(A_i)}^{s_i^{\max}}),$ where $s_i^{\max} = \max(S_i)$;
  else
    $X_{i+1} \leftarrow X_i^+$;
  end if;
end if.

Fig. 3. The behavior of M' in the proof of Theorem 8.

7 Conclusion

We introduced a new model of language learning called *temporary example memory* (Tem) learning. This model is a natural *restriction* of bounded example memory (Bem) learning. In particular, it requires that, if a learner commits an example x to memory in some stage of the learning process, then there is some subsequent stage for which x *no longer* appears in the learner's memory. In some sense, this model captures the idea that *memories fade*.

Aside from the open questions mentioned in Sections 5 and 6, the following would constitute an interesting line of research. In some sense, an IIM can memorize examples that it has seen by *coding* them into its hypotheses, i.e., by exploiting redundancy in the hypothesis space. This "memory" is, in principle, unbounded in the number of examples that it can retain, and in how long it can retain them.[7] From a practical point of view, the option to memorize examples in this way probably does not meet the *intuitive* requirements of a model of incremental learning. Thus, it would be interesting to consider the Bem and Tem-learning models in conjunction with hypothesis spaces *that have no redundancy*, i.e., Friedberg numberings. Note that such numberings have already been considered as hypothesis spaces in the context of It-learning [JS07].

[7] Of course, since the IIM must eventually converge to a single hypothesis, the IIM can memorize examples in this way only finitely often.

References

[Bak02] B. Bakker. Reinforcement learning with long short-term memory. *Advances in Neural Inform. Processing Systems*, 14:1475–1482, 2002.

[Cas74] J. Case. Periodicity in generations of automata. *Mathematical Systems Theory*, 8(1):15–32, 1974.

[Cas94] J. Case. Infinitary self-reference in learning theory. *Journal of Experimental and Theoretical Artificial Intelligence*, 6(1):3–16, 1994.

[CCJS07] L. Carlucci, J. Case, S. Jain, and F. Stephan. Results on memory-limited U-shaped learning. *Inform. Comput.*, 205(10):1551–1573, 2007.

[CJLZ99] J. Case, S. Jain, S. Lange, and T. Zeugmann. Incremental concept learning for bounded data mining. *Inform. Comput.*, 152(1):74–110, 1999.

[Gol67] E.M. Gold. Language identification in the limit. *Inform. Control*, 10(5):447–474, 1967.

[HMU01] J.E. Hopcroft, R. Motwani, and J.D. Ullman. *Introduction to automata theory, languages, and computation.* Addison Wesley, second edition, 2001.

[HS97] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[JS07] S. Jain and F. Stephan. Learning in Friedberg numberings. In *Proc. of 18th Intl. Conf. on Algorithmic Learning Theory*, volume 4754 of *LNCS*, pages 79–93, 2007.

[KS95] E. Kinber and F. Stephan. Language learning from texts: mind changes, limited memory, and monotonicity. *Inform. Comput.*, 123(2):224–241, 1995.

[LM92] L.J. Lin and T. Mitchell. Reinforcement learning with hidden states. In *Proc. of 2nd Intl. Conf. on Simulation of Adaptive Behavior*, pages 271–280, 1992.

[LMZ08] S. Lange, S.E. Moelius, and S. Zilles. Learning with temporary memory (expanded version). Technical report, University of Delaware, 2008. Available online at `http://www.cis.udel.edu/~moelius/publications`.

[LZ96] S. Lange and T. Zeugmann. Incremental learning from positive data. *Journal of Computer and System Sciences*, 53(1):88–103, 1996.

[LZZ08] Steffen Lange, Thomas Zeugmann, and Sandra Zilles. Learning indexed families of recursive languages from positive data: A survey. *Theor. Comput. Sci.*, 397(1-3):194–232, 2008.

[McC96] R.A. McCallum. Learning to use selective attention and short-term memory in sequential tasks. In *Proc. of 4th Intl. Conf. on Simulation of Adaptive Behavior*, pages 315–324, 1996.

[Mit97] T.M. Mitchell. *Machine Learning.* McGraw-Hill Higher Education, 1997.

[OSW86] D. Osherson, M. Stob, and S. Weinstein. *Systems that Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists.* MIT Press, Cambridge, Mass., first edition, 1986.

[RCN03] J.M. Rabaey, A. Chandrakasan, and B. Nikolic. *Digital Integrated Circuits: A Design Perspective.* Prentice-Hall, Inc., second edition, 2003.

[Rog67] H. Rogers. *Theory of Recursive Functions and Effective Computability.* McGraw Hill, New York, 1967. Reprinted, MIT Press, 1987.

[WC80] K. Wexler and P.W. Culicover. *Formal Principles of Language Acquisition.* MIT Press, Cambridge, Mass., 1980.

[Wie76] R. Wiehagen. Limes-Erkennung rekursiver Funktionen durch spezielle Strategien. *Elektron. Inform. Kybernetik*, 12(1/2):93–99, 1976.