Non U-Shaped Vacillatory and Team Learning [4]

Lorenzo Carlucci [a,1], John Case [b,1], Sanjay Jain [c,2], Frank Stephan [d,3]

[a] *Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716-2586,USA and Dipartimento di Matematica, Universitá di Siena, Pian dei Mantellini 44, Siena, Italy*

[b] *Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716-2586,USA*

[c] *School of Computing, National University of Singapore, 3 Science Drive 2, Singapore 117543, Republic of Singapore*

[d] *School of Computing and Department of Mathematics, National University of Singapore, 3 Science Drive 2, Singapore 117543, Republic of Singapore*

Abstract

U-shaped learning behaviour in cognitive development involves learning, unlearning and relearning. It occurs, for example, in learning irregular verbs. The prior cognitive science literature is occupied with how humans do it, for example, general rules versus tables of exceptions. This paper is mostly concerned with whether U-shaped learning behaviour may be *necessary* in the abstract mathematical setting of inductive inference, that is, in the computational learning theory following the framework of Gold. All notions considered are learning from text, that is, from positive data. Previous work showed that U-shaped learning behaviour is necessary for behaviourally correct learning but not for syntactically convergent, learning in the limit (= explanatory learning). The present paper establishes the necessity for the hierarchy of classes of vacillatory learning where a behaviourally correct learner has to satisfy the additional constraint that it vacillates in the limit between at most b grammars, where $b \in \{2, 3, \ldots, *\}$. Non U-shaped vacillatory learning is shown to be restrictive: every non U-shaped vacillatorily learnable class is already learnable in the limit. Furthermore, if vacillatory learning with the parameter $b = 2$ is possible then non U-shaped behaviourally correct learning is also possible. But for $b = 3$, surprisingly, there is a class witnessing that this implication fails.

1 Introduction and Motivation

U-shaped learning is a learning behaviour in which the learner first learns the correct behaviour, then abandons the correct behaviour and finally returns to the correct behaviour once again. This pattern of learning behaviour has been observed by cognitive and developmental psychologists in a variety of child development phenomena, such as language learning [9,28,41] understanding of temperature [41,42], understanding of weight conservation [8,41], object permanence [8,41] and face recognition [10]. The case of language acquisition is paradigmatic. In the case of the past tense of english verbs, it has been observed that children learn correct syntactic forms (call/called, go/went), then undergo a period of overregularization in which they attach regular verb endings such as 'ed' to the present tense forms even in the case of irregular verbs (break/breaked, speak/speaked) and finally reach a final phase in which they correctly handle both regular and irregular verbs. This example of U-shaped learning behaviour has figured so prominently in the so-called "Past Tense Debate" in cognitive science that competing models of human learning are often judged on their capacity for modeling the U-shaped learning phenomenon [28,35,43].

The prior literature is typically concerned with modeling *how* humans achieve U-shaped behaviour, while, in the present paper, we are mostly interested in *why* humans exhibit this seemingly inefficient behaviour. Is it a mere harmless evolutionary accident or is it *necessary* for full human learning power? Specifically, are there some learning tasks for which U-shaped behaviour is logically necessary? In the present paper we present some new theorems in the context of Gold's formal model of language learning from positive data [22] that suggest an answer to this latter question. To explain our results, we informally review the main notions of inductive inference, that is Gold-style learning theory, and refer to the next section for precise mathematical definitions.

A "learner" is modelled by a machine (algorithmic device) **M**. A "language" can be seen as a set of sentences and sentences are finite objects built from a finite alphabet. Thus one can use standard coding techniques (see [36]) and treat languages as subsets of the natural numbers. When learning a language L, the machine **M** reads, element by element, an infinite sequence $p_0, p_1, \ldots$ consisting of the elements of the language L in arbitrary order with possibly

Email addresses: `carlucci5@unisi.it` (Lorenzo Carlucci), `case@cis.udel.edu` (John Case), `sanjay@comp.nus.edu.sg` (Sanjay Jain), `fstephan@comp.nus.edu.sg` (Frank Stephan).
[1] Supported in part by NSF grant number NSF CCR-0208616.
[2] Supported in part by NUS grant number R252–000–127–112.
[3] Supported in part by NUS grant number R252–000–212–112.
[4] Some results of this and related work were presented at the 12th International Congress of Logic, Methodology and Philosophy of Science in Oviedo, Spain, 2003 and the 37th Annual Meeting of the Society for Mathematical Psychology in Ann Arbor, Michigan, 2004.

some pause symbols (denoted by "#") in between. Any such presentation of a language is called a "text" for the language. During this process the learner outputs a corresponding sequence $g_0\, g_1\, \ldots$ of hypotheses. These hypotheses model "grammars". A "grammar" can be seen as a finite set of rules describing a procedure for generating/producing/enumerating all and only the elements of a language L. A "grammar for/generating a language L" is taken to be such a procedure. We also say in this case that the grammar is "correct for L". In the setting of Gold-style learning theory, this procedure is taken to be algorithmic and thus can be thought of as a computer program. Any computer program can be coded using standard coding techniques (see [36]) as a natural number, so we assume that the output $g_0, g_1, \ldots$ of the machine $\mathbf{M}$ consists of codes for grammars which may generate the language L to be learned. We will use harmless ambiguity and forget the difference between a grammar and its code. These hypotheses, especially when numerically coded, are also called *indices*. In Gold-style learning theory we are exclusively concerned with learners learning *classes* of languages instead of single languages. This makes Gold-style learning theory a good framework for studying the problem of "natural languages" (since any proposed definition of "natural language" determines a class of languages fulfilling the definition).

A learner is said to learn a class $\mathcal{C}$ of languages if the learner learns each language in $\mathcal{C}$. What does it mean for a learner to learn a language? Gold-style learning theory features many different "learning criteria", that is, ways of saying that a learner successfully learns a language. We will here consider the three most prominent examples: *behaviourally correct learning*, *vacillatory learning* and *explanatory learning*. In *all* three cases we require the following, for a learner $\mathbf{M}$ to learn a language L: for *any* text for the language L, that is, for any presentation of the elements of L, after some point in the sequence of hypotheses $g_0, g_1, \ldots$ output by $\mathbf{M}$ only grammars that are correct for L appear. When this happens, the learner $\mathbf{M}$ is said to "converge" to a set of correct grammars for L. In this sense Gold-style learning theory is said to be a theory of "learning in the limit". The learning criteria (regarding successful learning of L) considered herein can each be defined by constraining the number of correct grammars to which the learner converges. Observe here that each language can be generated by many syntactically different grammars which are extensionally equivalent (in a computational setting, each language can be generated by infinitely many syntactically different grammars).

The concept of *behaviourally correct learning* puts no restriction on the number of correct grammars in the limit: a learner is said to "behaviourally learn/identify" a language L, if *almost all* grammars g_n generate the language L, that is, if the learner converges to a (possibly infinite) set of correct grammars for L. We use the acronym **TxtBc** (behaviourally correct learning from text) both to refer to the learning criterion and to the class of classes of languages that can be learned in this sense by some learner $\mathbf{M}$.

We now describe *vacillatory learning*. For $b \in \{1, 2, \ldots, \} \cup \{*\}$, the criterion $\mathbf{TxtFex}_b$ (finite explanatory learning from text, with bound b) requires that the g_n eventually vacillate between *at most* b correct grammars. This requirement says that there is a set of at most b correct grammars such that the learner outputs from some time on only grammars from this set (where "a set of at most $*$ correct grammars" stands for "a finite set of correct grammars"). So for each b we get (in principle) a different criterion/class of vacillatory learning. As above, the acronym $\mathbf{TxtFex}_b$ is used to refer both to the learning criterion and to the class of classes of languages learnable in this sense by some learner $\mathbf{M}$.

Explanatory learning requires that, on each text for L, the learner converges to *a single* correct grammar for L, that is, we require *syntactic* convergence. We use the acronym **TxtEx** (explanatory learning from text) to refer to the criterion and to the class of classes of languages that are learned in the explanatory sense by some learner $\mathbf{M}$. Observe that, by definition, **TxtEx** coincides with $\mathbf{TxtFex}_1$. We also use the locution "**TxtBc**- (respectively, $\mathbf{TxtFex}_b$-, **TxtEx**-) learner" for a class $\mathcal{C}$/language L, to indicate a learner learning $\mathcal{C}/L$ according to the criterion **TxtBc** (respectively $\mathbf{TxtFex}_b, \mathbf{TxtEx}$).

The basic relations between the criteria $\mathbf{TxtBc}, \mathbf{TxtFex}_b$ and **TxtEx** are as follows: more classes of languages are learnable in the **TxtBc** sense than in the $\mathbf{TxtFex}_b$ sense, for all $b \geq 1$. So behaviourally correct learning is said to be a more *powerful* learning criterion than vacillatory and exaplanatory learning. Also, as shown by Case in [11], for each $b \geq 1$, more classes are learnable by vacillating between at most $b + 1$ grammars than by vacillating between at most b grammars, so that $\mathbf{TxtFex}_{b+1}$ contains, as a class, more classes of languages than $\mathbf{TxtFex}_b$, for each $b \geq 1$. In this sense we say that the criteria $\mathbf{TxtFex}_1, \mathbf{TxtFex}_2, \ldots$ form a strict *hierarchy* of more and more *powerful* learning criteria. This hierarchy is called the *vacillatory hierarchy.*

We now come to the definition of U-shaped behaviour in our setting. A learner $\mathbf{M}$ is said to be *non U-shaped* on a language L if $\mathbf{M}$ learns the language L according to one of the learning criteria described above and, on all texts for L, g_{n+1} generates L whenever g_n does, in other words $\mathbf{M}$ never abandons correct conjecture. A learner is *non U-shaped* on a class if it is *non U-shaped* on each language in the class. A learner is *U-shaped* on a class (or on a language) if it is not non U-shaped on the class (or on the language). The present paper is concerned with the effect on the classes $\mathbf{TxtBc}, \mathbf{TxtFex}_b, \mathbf{TxtEx}$, learnable according to the three learning criteria described above, of *forbidding* U-shaped behaviour. We will be interested in the question: are all languages learnable in the **TxtBc** (respectively $\mathbf{TxtFex}_b, \mathbf{TxtEx}$) sense also learnable by some non U-shaped learner?

Baliga, Case, Merkle, Stephan and Wiehagen [4] initiated the Gold style learning theoretic study of U-shaped learning behaviour and showed that U-shaped learning is circumventable for explanatory learning, in the sense that every class in **TxtEx** can be learned by a learner which is non U-shaped on that class (see Theorem 9). In contrast to this, Fulk, Jain and Osherson [20, Proof of Theorem 4] showed that U-shaped learning behaviour is necessary for the "full learning power" of behaviourally correct learning, in the sense that there are some classes of languages in **TxtBc** such that any learner **TxtBc**-learning these classes is necessarily U-shaped on them. We show in Theorem 11 below that U-shaped learning behaviour is also necessary for full learning power for the whole hierarchy of vacillatory learning criteria, except for the base case of explanatory learning. Theorem 11 of the present paper shows that *non* U-shaped $\mathbf{TxtFex}_b$-learners are not more powerful than **TxtEx**-learners: the hierarchy collapses to $\mathbf{TxtFex}_1$ if U-shaped behaviour is forbidden. In other words, there are classes of languages that can be $\mathbf{TxtFex}_n$-identified, for $n > 1$, but these learners *must* be U-shaped on some text for some language in the class. What if we consider the more liberal criterion **TxtBc**? Our Theorem 22 strengthens the collapse result of Theorem 11 considerably by showing that there are classes in $\mathbf{TxtFex}_3$ (and therefore in $\mathbf{TxtFex}_4, \mathbf{TxtFex}_5, \ldots, \mathbf{TxtFex}_*$) such that there is no non U-shaped learner learning those classes in the **TxtBc**-sense. This means that U-shaped learning behaviour *cannot* be dispensed with for learning such classes, even if we only require behavioural convergence, allowing convergence to possibly infinitely many syntactically different correct hypotheses. By contrast, our last main result, Theorem 21, shows that every class of languages that can be learned in the $\mathbf{TxtFex}_2$ sense can be learned in the **TxtBc** sense by a non U-shaped learner. Hence, for only this early stage of the hierarchy, the cases in which $\mathbf{TxtFex}_2$-learning necessitates U-shaped learning behaviour can be circumvented by shifting to **TxtBc**-learning.

In Section 2 we introduce the notation and the basic definitions for the rest of the paper. We also include the basic results from [4] which are relevant for the present paper. In Section 3 we show that the class of classes of languages that are learnable in the $\mathbf{TxtFex}_b$ sense by some non U-shaped learner (we denote this class by $\mathbf{NUShTxtFex}_b$) actually coincides with **TxtEx** so that on one hand every **TxtEx**-learner can be simulated by a non U-shaped one while on the other hand vacillatory non U-shaped learners do not have more power than non U-shaped **TxtEx**-learners; that is, $\mathbf{NUShTxtFex}_b = \mathbf{TxtEx}$ for all $b \in \{1, 2, 3, \ldots, *\}$. In the subsequent sections we investigate the question whether one can obtain a non U-shaped learner from a $\mathbf{TxtFex}_b$ learner if one is willing to give up some constraints on the number of hypotheses considered. The positive result that all classes in $\mathbf{TxtFex}_2$ are learnable in the **TxtBc** sense by some non U-shaped learner is presented in Section 4 and the negative result that some classes in $\mathbf{TxtFex}_3$ are not learnable in the **TxtBc** sense by some non U-shaped learner is presented in Section 5. Section 6, as well as some previous sections, provides some results on non U-shaped team learning. In Section 7 we summarize our main results and provide a brief discussion on their relevance to U-shaped learning behaviour in cognitive science.

2 Preliminaries

$\mathbb{N}$ denotes the set of natural numbers, {0,1,2,...}. Unless otherwise specified the variables a, c, d, e, i, j, k range over $\mathbb{N}$. D, P, S range over *finite* sets of natural numbers. The cardinality function is denoted by card$(\cdot)$. card$(D) \leq *$ means that card(D) is finite. The symbol $*$ is used to denote the 'finite with no preassigned bound'. Unless otherwise stated, b will range over $\mathbb{N} \cup \{*\}$. The symbols $\subseteq, \subset, \supseteq, \supset$ respectively denote the subset, superset, proper subset and proper superset relation between sets. The union $\cup$ and the intersection $\cap$ are defined as usual; $\in$ denotes set-theoretic membership, $-$ denotes set-theoretic difference and $\mathbf{\Delta}$ denotes the symmetric difference of sets, that is, $A \mathbf{\Delta} B = (A \cup B) - (A \cap B)$. The quantifiers $\forall^\infty$ and $\exists^\infty$ mean 'for all but finitely many' and 'there exists infinitely many', respectively.

A pair $\langle i, j \rangle$ stands for an arbitrary, computable one-to-one encoding of all pairs of natural numbers onto $\mathbb{N}$ [36]. Similarly we can define $\langle \cdot, \ldots, \cdot \rangle$ for encoding n-tuples of natural numbers, for $n > 1$, onto $\mathbb{N}$.

φ denotes a fixed *acceptable* programming system for the partial computable functions [36]. φ_e denotes the partial computable function computed by the program with code number e in the φ-system. We will unambiguously refer to programs using their code number in the φ-system. We let H, I, J, L range over recursively enumerable sets and $\mathcal{C}, \mathcal{L}$ range over classes of recursively enumerable sets. $K = \{e : e \in W_e\}$, the diagonal halting problem, is a standard example for a nonrecursive r.e. set.

Furthermore, we fix a uniformly recursive enumeration of all r.e. sets such that

- W_e is the domain of φ_e and $W_e = \bigcup_s W_{e,s}$;
- $W_{e,0} = \emptyset$ and $W_{e,s} \subseteq W_{e,s+1} \subseteq \{0, \ldots, s\}$ for all e, s;
- $\{(e, s, x) : x \in W_{e,s}\}$ is recursive;
- $\{(x, s) : x \in W_{e,s}\}$ is primitive recursive for all e;
- For every primitive-recursive enumeration A_s of some set A with $A_0 = \emptyset$ $\wedge (\forall s)\, [A_s \subseteq A_{s+1} \subseteq \{0, \ldots, s\}]$ there is an index e with $(\forall s)\, [W_{e,s} = A_s]$; furthermore, e can be computed from an index of the enumeration for A_s.

Of course, $W_{e,s}$ can be easily based on a specifically designed Blum complexity measure from [7]. Any unexplained recursion-theoretic notions are from [30,36].

We now introduce the basic definitions of Gold-style computational learning theory.

Definition 1 A *sequence* σ is a mapping from an initial segment of $\mathbb{N}$ into $\mathbb{N} \cup \{\#\}$. An *infinite sequence* is a mapping from $\mathbb{N}$ into $\mathbb{N} \cup \{\#\}$. The content of a finite or infinite sequence σ is the set of natural numbers occurring in σ and is denoted by content(σ). The length of a sequence σ is the number of elements in the domain of σ and is denoted by $|\sigma|$. For a subset L of $\mathbb{N}$, seg(L) denotes the set of sequences σ with content$(\sigma) \subseteq L$. An infinite sequence T is a *text for* L iff $L =$ content(T).

The symbol # is mainly introduced to uniformly deal with the empty language and intuitively represents a pause in the presentation of the language to the learner. Concatenation of two sequences σ and τ is denoted by $\sigma\tau$. If $x \in$

$(\mathbb{N} \cup \{\#\})$, then σx means $\sigma\tau$ where τ is the sequence consisting of exactly one element which is x. $\sigma \subseteq \tau$ means that σ is an initial segment of τ and $\sigma \subset \tau$ means that σ is a proper initial segment of τ.

Intuitively, a *text* for a language L is an infinite stream or sequential presentation of *all* the elements of the language L in any order and with the #'s representing pauses in the presentation of the data. For example, the only text for the empty language is an infinite sequence of #'s. Technically, a text is a mapping from $\mathbb{N}$ into $(\mathbb{N} \cup \{\#\})$. We let T, with possible subscripts and superscripts, range over texts. $T[n]$ denotes the finite initial segment of T with length n. Observe that the domain of $T[n]$ is $\{x : x < n\}$. So $T(n)$ is not a member of the sequence $T[n]$. $\sigma \subset T$ denotes the fact that σ is an initial segment of T. Observe that in this case we have $\sigma = T[|\sigma|]$.

A learner will map sequences to hypotheses. These are represented by natural numbers and interpreted as codes for programs in the φ-system. **M**, with possible superscripts and subscripts, is intended to range over language learning machines.

Definition 2 [4,5,11,14,15,22,33] A *language learning machine* **M** is a computable mapping from seg($\mathbb{N}$) into $\mathbb{N}$. **M** **TxtBc**-learns a class $\mathcal{L}$ of r.e. languages iff for every $L \in \mathcal{L}$ and every text T for L, almost all hypotheses $\mathbf{M}(T[n])$ are indices for the language L to be learned.

A **TxtBc**-learner **M** for $\mathcal{L}$ is a $\mathbf{TxtFex}_*$-learner for $\mathcal{L}$ iff for every $L \in \mathcal{L}$ and every text T for L the set $\{T[n] : n \in \mathbb{N}\}$ is finite.

A $\mathbf{TxtFex}_*$-learner **M** for $\mathcal{L}$ is a $\mathbf{TxtFex}_b$-learner for $\mathcal{L}$, where $b \in \{1, 2, \ldots\}$, iff there are for every $L \in \mathcal{L}$ and every text T for L at most b indices which **M** outputs infinitely often, that is, $|\{e : (\exists^\infty n)\, [e = \mathbf{M}(T[n])]\}| \leq b$.

A **TxtBc**-learner **M** for $\mathcal{L}$ is a **TxtEx**-learner for $\mathcal{L}$ iff for every $L \in \mathcal{L}$ and every text T for L almost all hypotheses $\mathbf{M}(T[n])$ are the same grammar for L.

A **TxtBc**-learner **M** for $\mathcal{L}$ is non U-shaped iff for every $L \in \mathcal{L}$ and every text T for L there are no three numbers k, m, n such that $k < m < n$ and $W_{\mathbf{M}(T[k])} = L$, $W_{\mathbf{M}(T[m])} \neq L$ and $W_{\mathbf{M}(T[n])} = L$. Furthermore, $\mathbf{NUShTxtFex}_b$-learners and **NUShTxtEx**-learners (of a class $\mathcal{L}$) are those learners which are non U-shaped and at the same time a $\mathbf{TxtFex}_b$-learner and **TxtEx**-learner, respectively, for $\mathcal{L}$.

The criteria **TxtBc**, $\mathbf{TxtFex}_b$, **TxtEx**, **NUShTxtBc**, $\mathbf{NUShTxtFex}_b$, **NUShTxtEx** are the sets consisting of all those classes which are learnable by a learner satisfying the respective above defined requirements.

The part "**Txt**" in the acronyms relates to the fact that all notions considered in the present work are learning from text, that is, from positive data. Intuitively, the notion **TxtBc** captures what could be called learning in the most general sense. All other notions are restrictions. The historically most important one is the notion **TxtEx** introduced by Gold [22] where the learner has to converge syntactically to a single index of the language to be learned.

Intuitively a class $\mathcal{L}$ of r.e. languages is $\mathbf{TxtFex}_b$ identified by a machine $\mathbf{M}$ iff when $\mathbf{M}$ is given as input *any* listing T of *any* $L \in \mathcal{L}$, it outputs a sequence of grammars such that, past some point in this sequence, no more than b *syntactically* different grammars occur and each of them is a grammar for L. The acronym **TxtFex** stands for 'finite explanatory identification from text'. $\mathbf{TxtFex}_1$ is equivalent to Gold's original notion of *identification*, also denoted by **TxtEx** and called 'explanatory identification from text'. Osherson and Weinstein [31] first studied $\mathbf{TxtFex}_*$-identification, later Case [11] studied the whole hierarchy with $b \in \{1, 2, \ldots\}$.

Definition 3 (a) [19] We say that σ is a **TxtEx**-*stabilizing sequence* for a learner $\mathbf{M}$ on a set L iff $\sigma \in \text{seg}(L)$ and $\mathbf{M}(\sigma\tau) = \mathbf{M}(\sigma)$ for all $\tau \in \text{seg}(L)$.

(b) [6] σ is called a **TxtEx**-*locking sequence* for $\mathbf{M}$ on L iff σ is a stabilizing sequence for $\mathbf{M}$ on L and $W_{\mathbf{M}(\sigma)} = L$.

Lemma 4 [6] *Suppose* $\mathbf{M}$ **TxtEx**-*identifies* L. *Then,*

(a) *there exists a* **TxtEx**-*locking sequence for* $\mathbf{M}$ *on* L*;*

(b) *for every* $\sigma \in seg(L)$*, there exists a* $\tau \in seg(L)$*, such that* $\sigma\tau$ *is a* **TxtEx**-*locking sequence for* $\mathbf{M}$ *on* L*;*

(c) *every* **TxtEx**-*stabilizing sequence* σ *for* $\mathbf{M}$ *on* L*, is also a* **TxtEx**-*locking sequence for* $\mathbf{M}$ *on* L*.*

Note that the definitions for stabilizing and locking sequence, as well as Lemma 4, can be generalized to other learning criteria such as $\mathbf{TxtFex}_b$ and **TxtBc**. We often omit the term like "**TxtEx**" from **TxtEx**-locking sequence, when it is clear from context.

Smith [38,39] studied learning by teams of machines. We show that vacillatory learning can be characterized by teams as below. Before that we recall the definition of team learning.

Definition 5 [38] A class $\mathcal{L}$ is in $[a, b]$**TxtEx** iff there is a team, $\mathbf{M}_1, \mathbf{M}_2, \ldots,$ $\mathbf{M}_b$, of b machines such that on any text for any language $L \in \mathcal{L}$, at least a of the b machines in the team converge to an index for L.

Theorem 6 *A class* $\mathcal{L}$ *has a* $\mathbf{TxtFex}_b$-*learner iff there is a team of* b *machines* $\mathbf{N}_1, \ldots, \mathbf{N}_b$ *such that for every* $L \in \mathcal{L}$ *and for every text* T *for* L*, each machine* $\mathbf{N}_a$ *converges to a single index* e_a *and at least one of these indices* e_a *is an index for* L*.*

Proof. If $\mathcal{L}$ is $\mathbf{TxtFex}_b$ learnable by $\mathbf{M}$ and $L \in \mathcal{L}$ and T is a text for L, then one can define b new machines $\mathbf{N}_1, \ldots, \mathbf{N}_b$ such that at every input σ one computes a list $e_1, \ldots, e_b$ of the most recent conjectures of $\mathbf{M}$ ordered by size and each machine $\mathbf{N}_a$ outputs e_a; in the case that there are less than b hypotheses in total, the list is made to have size b by using arbitrary constants. Since $\mathbf{M}$

eventually vacillates between at most b hypothesis on T, the considered list $e_1, \ldots, e_b$ stabilizes after finite time and from then on all machines $\mathbf{N}_a$ do not change their hypothesis. Furthermore, since $\mathbf{M}$ vacillates eventually between b or fewer correct indices, some machine $\mathbf{N}_a$ converges to a correct hypothesis.

For the other way round, consider that $\mathbf{N}_1, \ldots, \mathbf{N}_b$ is a team of machines such that on every text T for any language $L \in \mathcal{L}$, each $\mathbf{N}_a$ converges to some index and one of these indices is correct. Then $\mathbf{M}$ can be defined such that $\mathbf{M}(T[n])$ takes one of those hypotheses $\mathbf{N}_a(T[n])$ for which the quality of the hypothesis measured as

$$q_{T[n],a} = \max(\{x \leq n : (\forall y < x)\,[y \in W_{\mathbf{N}_a(T[n]),n} \Leftrightarrow y \in \text{content}(T[n])\,]\})$$

is maximal. By hypothesis, all b machines converge to an index. Thus the learner $\mathbf{M}$ eventually vacillates between at most b hypotheses. Furthermore, if $\mathbf{N}_a$ converges on T to an index e_a of L then $q_{T[n],a}$ converges to ∞; if $\mathbf{N}_a$ converges on T to an index e_a of a set different from L then $q_{T[n],a}$ converges to a finite number in $\mathbb{N}$. Thus, from some time on, $\mathbf{M}$ will only consider correct indices and thus $\mathbf{M}$ $\mathbf{TxtFex}_b$-learns $\mathcal{L}$. ∎

Case [11] showed that the criteria $\mathbf{TxtFex}_b$, $b = 1, 2, 3, \ldots, *$, form a hierarchy of more and more powerful learning criteria, that is, $\mathbf{TxtFex}_1 \subset \mathbf{TxtFex}_2 \subset \ldots \subset \mathbf{TxtFex}_*$, as stated in the following Theorem.

Theorem 7 [11] *For each $b \in \{1, 2, \ldots\}$, the separation $\mathbf{TxtFex}_{b+1} \not\subseteq \mathbf{TxtFex}_b$ is witnessed by the self-referential class*

$$\{W_e : e \in W_e \wedge |W_e \cap \{0, \ldots, e\}| \leq b + 1\}$$

consisting of all nonempty sets which have an index among its smallest $b+1$ elements. Furthermore consider the class

$$\{W_e : W_e \neq \emptyset \wedge e \leq \min(W_e)\}$$

consisting of all nonempty sets where the smallest index is bounded by their smallest element. This class witnesses that $\mathbf{TxtFex}_* \not\subseteq \mathbf{TxtFex}_b$ for any $b \in \{1, 2, \ldots\}$.

Proposition 8 $\mathbf{NUShTxtBc} \not\subseteq \mathbf{TxtFex}_*$.

Proof. Let $K = \{e : e \in W_e\}$ and $\mathcal{K} = \{K \cup E : E \text{ is finite}\}$. We will show that $\mathcal{K} \in \mathbf{NUShTxtBc} - \mathbf{TxtFex}_*$. The $\mathbf{NUShTxtBc}$-learner is given by a machine which outputs an hypothesis for $K \cup \text{content}(\sigma)$ for every input σ. It is easy to see that for every finite set E and every text T for $K \cup E$, after the learner has seen all the elements of E, it outputs hypotheses for $K \cup E$. Furthermore, the learner is strongly monotone, that is, $W_{\mathbf{M}(\sigma)} = K \cup \text{content}(\sigma) \subseteq K \cup \text{content}(\sigma\tau) = W_{\mathbf{M}(\sigma\tau)}$ for all $\sigma, \tau \in \text{seg}(\mathbb{N})$. Thus, $\mathbf{M}$ $\mathbf{NUShTxtBc}$-identifies $\mathcal{K}$.

On the other hand, suppose by way of contradiction that $\mathbf{M}$ $\mathbf{TxtFex}_*$-identifies $\mathcal{K}$. Then (using an analogue of Lemma 4) there is a $\mathbf{TxtFex}_*$-locking sequence σ for $\mathbf{M}$ on K in the sense that $\sigma \in \mathrm{seg}(K)$ and for some finite set F of indices of K, $\mathbf{M}(\sigma\tau) \in F$ for all $\tau \in \mathrm{seg}(K)$.

Now, if $x \notin K$, then there is a sequence $\tau \in \mathrm{seg}(K)$ such that $\mathbf{M}(\sigma x \tau) \notin F$ since no index in F is an index for $K \cup \{x\}$. Thus, having F and σ, the following Π^0_1-predicate defines the set K:

$$x \in K \Leftrightarrow (\forall \tau \in \mathrm{seg}(K))\, [\mathbf{M}(\sigma x \tau) \in F].$$

But, then, K would be recursive. By this contradiction, $\mathcal{K}$ is not $\mathbf{TxtFex}_*$-learnable. ∎

The next theorem is from [4] and states that being non U-shaped is not restrictive for **TxtEx**-learning. Its proof can also be obtained by letting $b = 1$ in Theorem 16 below. This will be extended to $\mathbf{NUShTxtFex}_b = \mathbf{TxtEx}$ in Corollary 12.

Theorem 9 [4] $\mathbf{NUShTxtFex}_1 = \mathbf{TxtEx}$.

Hence, for **TxtEx**, U-shaped behaviour is not necessary for full learning power. That is, every class in **TxtEx** can be learned by a non U-shaped **TxtEx**-learner. By contrast, by an easy adaptation of the proof of Theorem 4 in [20], we have the following.

Theorem 10 [4,20] $\mathbf{NUShTxtBc} \subset \mathbf{TxtBc}$.

Hence, U-shaped behaviour cannot be avoided for achieving the full learning power of **TxtBc**. That is, there is a class $\mathcal{L}$ of languages in **TxtBc** such that every **TxtBc**-learner for $\mathcal{L}$ shows U-shaped behaviour on some text T for some language $L \in \mathcal{L}$.

3 U-shaped vacillatory learning

We first show, by a simple counting argument, that the hierarchy $\mathbf{TxtFex}_1 \subset \mathbf{TxtFex}_2 \subset \ldots \subset \mathbf{TxtFex}_*$ of vacillatory learning criteria collapses to $\mathbf{TxtFex}_1$ if U-shaped behaviour is forbidden.

Theorem 11 $\mathbf{NUShTxtFex}_* \subseteq \mathbf{TxtFex}_1$.

Proof. Let $\mathcal{L} \in \mathbf{NUShTxtFex}_*$ and let $\mathbf{M}$ be a learner witnessing this fact. We define a new learner $\mathbf{N}$ witnessing that $\mathcal{L} \in \mathbf{TxtFex}_1$ as follows. By padding, one can assume that if $\mathbf{M}(\tau) \notin \{\mathbf{M}(\sigma) : \sigma \subset \tau\}$, then $\mathbf{M}(\tau) > \mathbf{M}(\sigma)$, for all $\sigma \subset \tau$.

On input τ, $\mathbf{N}(\tau) = \max(\{\mathbf{M}(\sigma) : \sigma \subseteq \tau\})$.

Given $L \in \mathcal{L}$ and a text T for L, there is a least n such that $\mathbf{M}(T[n])$ is the maximal hypothesis which $\mathbf{M}$ outputs on T. This maximum exists, since $\mathbf{M}$ outputs only finitely many hypothesis. Note that $\mathbf{N}(T[m]) = \mathbf{M}(T[n])$ for all $m \geq n$.

Now any hypotheses of $\mathbf{M}$ on T has already been output on input $T[m]$, for some $m \leq n$ and for one of these m, this hypothesis is correct. So $\mathbf{M}(T[m])$ is an index for L and since $\mathbf{M}$ is non U-shaped, $\mathbf{M}(T[n])$ is also an index for L. Hence, $\mathbf{N}$ converges on T to the index $\mathbf{M}(T[n])$ for L. Thus, $\mathbf{N}$ is a $\mathbf{TxtFex}_1$-learner for $\mathcal{L}$. ∎

Theorems 9 and 11 give $\mathbf{NUShTxtFex}_1 = \mathbf{TxtFex}_1$ and $\mathbf{NUShTxtFex}_* \subseteq \mathbf{NUShTxtFex}_1$. Furthermore, the definition of $\mathbf{NUShTxtFex}_b$ immediately gives $\mathbf{NUShTxtFex}_1 \subseteq \mathbf{NUShTxtFex}_b \subseteq \mathbf{NUShTxtFex}_*$. Thus all these criteria coincide.

Corollary 12 $(\forall b \in \{1, 2, \ldots, *\})\, [\mathbf{NUShTxtFex}_b = \mathbf{TxtFex}_1]$.

The result $\mathbf{NUShTxtFex}_1 = \mathbf{TxtFex}_1$ stands in contrast to the fact that $\mathbf{TxtFex}_1 \subset \mathbf{TxtFex}_2 \subset \ldots \subset \mathbf{TxtFex}_*$. Thus we have that the following inclusions are proper.

Corollary 13 $(\forall b \in \{2, 3, \ldots, *\})\, [\mathbf{NUShTxtFex}_b \subset \mathbf{TxtFex}_b]$.

Corollaries 12 and 13 show that U-shaped learning behaviour is *necessary* for the full learning power of $\mathbf{TxtFex}_b$-identification for $b > 1$ in a strong sense: if U-shaped learning behaviour is forbidden, the hierarchy collapses to $\mathbf{TxtFex}_1$. Hence, for $b > 1$, the $\mathbf{TxtFex}_b$-learnability of any class in $(\mathbf{TxtFex}_b - \mathbf{TxtFex}_1)$ *requires* U-shaped learning behaviour. Recall from Theorem 7 that for any $b \in \{1, 2, \ldots\}$ the class

$$\mathcal{L}_{b+1} = \{W_e : e \in W_e \wedge |W_e \cap \{0, \ldots, e\}| \leq b + 1\}$$

is in $\mathbf{TxtFex}_{b+1} - \mathbf{TxtFex}_b$. This class is then the example in the following corollary which is an easy consequence of Theorem 7 and Corollary 12.

Corollary 14 *Let $b \in \{2, 3, \ldots\}$. Then any $\mathbf{M}$ witnessing $\mathcal{L}_b \in \mathbf{TxtFex}_b$ necessarily employs U-shaped learning behaviour on $\mathcal{L}_b$.*

A non U-shaped learner does not make a mind change from a correct to an incorrect hypothesis since it cannot learn the set otherwise. We require this property on all machines for the case of team learning. At the end of Section 7, we discuss an alternative approach.

Definition 15 A class $\mathcal{L}$ is in $[a, b]\mathbf{NUShTxtEx}$ iff there are b machines such that on any text for any language L in $\mathcal{L}$, (a) at least a machines in the team converge to an index for L and (b) no machine in the team makes a mind change from an index for L to an index for some other language.

The next result shows that Theorem 6 can be extended such that every class in $\mathbf{TxtFex}_b$ is learnable from a non U-shaped team. So the restriction $\mathbf{NUShTxtFex}_b = \mathbf{TxtFex}_1$ is caused by the fact that the hypothesis of the learner have to be brought into an ordering and cannot be done in parallel as in the case of the team below. Actually Theorem 16 enables us to achieve more properties of the team than that it is just non U-shaped.

Theorem 16 *Let $b \in \mathbb{N}^+$ and $\mathcal{L} \in$ **TxtFex**$_b$. Then there is a team of b learners $\mathbf{M}_1, \ldots, \mathbf{M}_b$ such that for all $L \in \mathcal{L}$ and all texts T for L there is an $n \in \mathbb{N}$ such that,*

(1) *$T[n]$ is a stabilizing sequence of all members of the team on L, in particular $\mathbf{M}_a(T[m]) = \mathbf{M}_a(T[n])$ for all $m \geq n$;*
(2) *there is an $a \in \{1, \ldots, b\}$ such that $\mathbf{M}_a(T[n])$ is an index for L;*
(3) *if $a \in \{1, \ldots, b\}$ and $\mathbf{M}_a(T[m])$ is an index for L then $m \geq n$.*

*In particular, $\mathbf{M}_1, \ldots, \mathbf{M}_b$ $[1, b]$***NUShTxtEx***-learns $\mathcal{L}$.*

Proof. By Theorem 6 there is a team $\mathbf{N}_1, \ldots, \mathbf{N}_b$ of **TxtEx**-learners for $\mathcal{L}$ such that for every $L \in \mathcal{L}$ and every text T for L, every machine converges on T to some hypothesis and at least one of these hypotheses is an index for L.

The basic idea of the proof is to search for a $\sigma \in \text{seg}(L)$, which is a **TxtEx**-stabilizing-sequence for each member of the team $\mathbf{N}_1, \ldots, \mathbf{N}_b$ on L. Additionally, we will also find a maximal set $D \subseteq \{1, \ldots, b\}$ such that σ is a stabilizing sequence for each $\mathbf{N}_i$, $i \in \{1, \ldots, b\}$ on $W_{\mathbf{N}_j(\sigma)}$, $j \in D$. Before such σ, D, is obtained, we will make sure that the output of $\mathbf{M}_a$ below is not a grammar for L. Once such σ, D is obtained, we will have that the learners $\mathbf{M}_a$ do not change their hypothesis and one of them correctly outputs a grammar for L. We now proceed formally.

Let E to be an infinite recursive set such that $E \cup \tilde{E} \notin \mathcal{L}$ for all finite sets $\tilde{E}$. Such an E can be defined as follows. If $\mathbb{N} \notin \mathcal{L}$, then let $E = \mathbb{N}$. If $\mathbb{N} \in \mathcal{L}$, then there exists a **TxtFex**$_b$-locking sequence τ for $\mathbf{M}$ on $\mathbb{N}$. Now we can take E to be any infinite and coinfinite recursive set such that $\text{content}(\tau) \subseteq E$.

Let $\sigma \sqsubseteq \tau$ denote that $\text{content}(\sigma) \subseteq \text{content}(\tau)$ and $|\sigma| \leq |\tau|$. Furthermore, let T_e be the canonical text for W_e, that is, T_e is the text generated by some standard enumeration of W_e.

As long as the content of the input is $\emptyset$ or no σ is found in the algorithm below, all machines $\mathbf{M}_1, \ldots, \mathbf{M}_b$ output the least index of $\emptyset$.

The σ searched for on input $T[t]$ has to satisfy the following conditions:

(a) $\sigma \sqsubseteq T[t]$ and $\text{content}(\sigma) \neq \emptyset$;
(b) $\mathbf{N}_a(\sigma\eta) = \mathbf{N}_a(\sigma)$ for all $a \in \{1, \ldots, b\}$ and $\eta \sqsubseteq T[t]$.

Once having σ, this is only replaced by a σ' on a future input $T[t']$ iff σ' but not σ satisfies (a) and (b) with respect to $T[t']$ (if there are several choices to replace σ, the first one with respect to some fixed recursive enumeration of $\text{seg}(\mathbb{N})$ is taken). Having σ, define D as follows.

(c) $D = \{a \in \{1, ..., b\} : (\forall \eta \sqsubseteq T_{\mathbf{N}_a(\sigma)}[t]) \, (\forall c \in \{1, ..., b\}) \, [\mathbf{N}_c(\sigma\eta) = \mathbf{N}_c(\sigma)]\}$.

Having σ and D, $\mathbf{M}_a(\tau) = F(\sigma, D, a)$ where $W_{F(\sigma,D,a)}$ is the set of all x for which there is an s such that the conditions (d) and either (e) or (f) below hold.

(d) $a \in D$ and $\sigma \sqsubseteq \text{content}(T_{\mathbf{N}_a(\sigma)}[s])$;

(e) $x \in \text{content}(T_{\mathbf{N}_a(\sigma)}[s])$ and $\mathbf{N}_c(\sigma\eta) = \mathbf{N}_c(\sigma)$ for all $c \in \{1, \ldots, b\}$, $d \in D$ and $\eta \sqsubseteq T_{\mathbf{N}_d(\sigma)}[s]$;
(f) $x \in E$ and $\mathbf{N}_c(\sigma\eta) \neq \mathbf{N}_c(\sigma)$ for some $c \in \{1, \ldots, b\}$, $d \in D$ and $\eta \sqsubseteq T_{\mathbf{N}_d(\sigma)}[s]$.

It is easy to see that the sets $W_{F(\sigma,D,a)}$ are uniformly recursively enumerable and thus the specified function F can be taken to be recursive. Thus also the learners $\mathbf{M}_1, \ldots, \mathbf{M}_b$ are recursive.

Now the properties (1), (2) and (3) are verified.

If $L = \emptyset$ then all machines $\mathbf{M}_1, \ldots, \mathbf{M}_b$ output on every prefix of every text for L the least index of $\emptyset$ and thus (1), (2) and (3) are satisfied with the parameter $n = 0$.

Now consider any $L \in \mathcal{L} - \{\emptyset\}$ and let T be a text for L. Let σ_t, D_t be the values of σ, D as chosen on input $T[t]$. It is easy to verify that if $\sigma_{t+1} = \sigma_t$, then $D_{t+1} \subseteq D_t$. Also note that if $\sigma_t \neq \sigma_{t+1}$, then $\sigma_t \neq \sigma_{t'}$, for all $t' > t$. Thus, one does not return to an abandoned σ.

(1) Since all machines $\mathbf{N}_1, \ldots, \mathbf{N}_b$ converge on every text for L, one can construct by induction sequences $\eta_1, \eta_2, \ldots, \eta_b \in \text{seg}(L)$ such that $\eta_1 \subseteq \eta_2 \subseteq \ldots \subseteq \eta_b$ and $\mathbf{N}_c(\eta_a) = \mathbf{N}_c(\eta_a\vartheta)$ for all $a \in \{1, \ldots, b\}$, $c \in \{1, \ldots, a\}$ and $\vartheta \in \text{seg}(L)$. Since η_b satisfies (a) and (b) with respect to input $T[t]$ for all sufficiently long t, it follows that $\sigma_{fl} = \lim_{t\to\infty} \sigma_t$ is defined and hence $D_{fl} = \lim_{t\to\infty} D_t$ is also defined. Let n be the least number such that $\sigma_n = \sigma_{fl}$ and $D_n = D_{fl}$. Now $\mathbf{M}_a(T[m]) = F(\sigma_{fl}, D_{fl}, a)$, for $a \in \{1, \ldots, b\}$ and $m \geq n$.

(2) Since σ_{fl} is a stabilizing sequence on L for $\mathbf{N}_1, \ldots, \mathbf{N}_b$, there is an index a with $\mathbf{N}_a(\sigma_{fl}) = L$. It follows from (c) that $a \in D_{fl}$. Furthermore, $\sigma_{fl} \sqsubseteq T_{\mathbf{N}_a(\sigma_{fl})}$ since $\sigma_{fl} \sqsubseteq T$ and $T, T_{\mathbf{N}_a(\sigma)}$ are both texts for the same language L. Now consider the definition of $\mathbf{M}_a(T[n]) = F(\sigma_n, D_n, a)$. For all x, (d) is satisfied for all sufficiently large s. If (f) would hold for some sufficiently large s and $d \in D_{fl}$ then d would not be in D_t for sufficiently large inputs $T[t]$ in contradiction to D_{fl} being the final value which D_t takes. So one can conclude that for every $x \in L$ the condition (e) is satisfied for all sufficiently large s and thus $W_{\mathbf{M}_a(T[n])} = L$.

(3) Assume that $\mathbf{N}_{a'}(T[m]) = L$. Since L is not empty, $a' \in D_m$. Note that $W_{F(\sigma_m,D_m,a')}$ is different from L if the case (f) in the definition of $W_{F(\sigma_m,D_m,a')}$ is satisfied for any $x \in E$ and $d \in D_m$: then $W_{F(\sigma_m,D_m,a')}$ is the union of E and a finite set and thus not in $\mathcal{L}$ by the choice of E. So, σ_m is a stabilizing sequence for $\mathbf{N}_1, \ldots, \mathbf{N}_b$, on any $W_{\mathbf{N}_d(\sigma_m)}$ with $d \in D_m$. Since L is among these $W_{\mathbf{N}_d(\sigma_m)}$, one can conclude that $\sigma_t = \sigma_m$ for all $t \geq m$ and thus $\sigma_m = \sigma_{fl}$. Furthermore, all members of D_m satisfy the selection condition (c) for all t and $D_m \subseteq D_{fl}$. Thus, $D_m = D_{fl}$ (since $\sigma_{fl} = \sigma_m$ implies $D_{fl} \subseteq D_m$). Since n is the first number where the parameters (σ_n, D_n) are equal to (σ_{fl}, D_{fl}), $m \geq n$.

So one has that every class in $\mathbf{TxtFex}_b$ is learnable by a team of b machines which abstains from any further hypothesis changes whenever one of the team members outputs a correct hypothesis. In particular the team $\mathbf{M}_1, \ldots, \mathbf{M}_b$ is non U-shaped. ∎

Theorem 17 $\mathbf{TxtFex}_b \subseteq [2, b+1]\mathbf{NUShTxtEx}$ *for all* $b \in \mathbb{N}^+$.

Proof. For a $\mathbf{TxtFex}_b$-learnable class $\mathcal{L}$, construct a team of learners $\mathbf{M}_1, \ldots, \mathbf{M}_b$ and the set E as in the proof of Theorem 16. A further learner $\mathbf{M}_{b+1}$ is added which combines the hypotheses of the other learners. Let T be a text for some language $L \in \mathcal{L}$. On input $T[m]$, $\mathbf{M}_{b+1}$ does the following where C_m denotes content$(T[m])$.

- Check whether there is an $x \leq m$ such that exactly one $a \in \{1, \ldots, b\}$ satisfies $W_{\mathbf{M}_a(T[m]),m} \cap \{0, \ldots, x\} = C_m \cap \{0, \ldots, x\}$.
- If not, output an index for E and halt.
- If so, let a_m be this unique a and x_m be the least such x.
- Compute from a_m, x_m an index k_m such that

$$W_{k_m} = \begin{cases} W_{\mathbf{M}_{a_m}(T[m])} & \text{if } W_{\mathbf{M}_c(T[m]),m}(y) = W_{\mathbf{M}_c(T[m])}(y) \\ & \text{for all } y \leq x_m \text{ and all } c \in \{1, \ldots, b\}; \\ E \cup \tilde{E} & \text{for some finite } \tilde{E}, \text{ otherwise.} \end{cases}$$

- Output the hypothesis k_m and halt.

It is easy to see that $\mathbf{M}_{b+1}$ is recursive. Now fix $L \in \mathcal{L}$ and let T be a text for L.

Now it is shown that the machine $\mathbf{M}_{b+1}$ and thus the whole team is non U-shaped on T. That is, $\mathbf{M}_{b+1}$ does not make a mind change from a correct to an incorrect hypothesis. Let m be given and assume that $\mathbf{M}_{b+1}(T[m])$ is correct, that is, $L = W_{\mathbf{M}_{b+1}(T[m])}$. Then $W_{\mathbf{M}_{b+1}(T[m])} \neq E \cup \tilde{E}$ for all finite $\tilde{E}$ by the choice of E in the proof of Theorem 16. Thus $W_{\mathbf{M}_{b+1}(T[m])} = W_{\mathbf{M}_{a_m}(T[m])}$ and $m \geq n$ for the n from Theorem 16. Furthermore, the following conditions hold for all $o \geq m$.

- $W_{\mathbf{M}_c(T[m]),m}(y) = W_{\mathbf{M}_c(T[m])}(y)$ for all $y \leq x_m$ and all $c \in \{1, \ldots, b\}$;
- $L \cap \{0, \ldots, x_m\} = C_m \cap \{0, \ldots, x_m\} = C_o \cap \{0, \ldots, x_m\}$;
- For all $c \in \{1, \ldots, b\} - \{a_m\}$ there is $y \leq x_m$ such that $W_{\mathbf{M}_{a_m},o}(y) = W_{\mathbf{M}_{a_m}(T[m]),m}(y) \neq W_{\mathbf{M}_c(T[m]),m}(y) = W_{\mathbf{M}_c(T[m]),o}(y)$;
- x_o, a_o exist and $x_o = x_m$, $a_o = a_m$, $k_o = k_m$.

So $\mathbf{M}_{b+1}(T[o]) = \mathbf{M}_{b+1}(T[m])$ for all $o \geq m$ and $\mathbf{M}$ does not make a mind change from a correct to an incorrect hypothesis.

It remains to show that the team $[2, b+1]\mathbf{TxtEx}$-learns L. From Theorem 16, one learner $\mathbf{M}_a$ with $a \in \{1, \ldots, b\}$ learns L. Assume now that $\mathbf{M}_a$ is the only one to do so; otherwise nothing has to be shown. Let m be so large that

- $m \geq n$ for the n from Theorem 16 and the machines $\mathbf{M}_1, \ldots, \mathbf{M}_b$ do not change their hypothesis beyond $T[m]$.

- $x \leq m$ for the least x such that for all $c \in \{1, \ldots, b\} - \{a\}$ there is an $y \leq x$ with $L(y) \neq W_{\mathbf{M}_c(T[m])}(y)$;
- for all $c \in \{1, \ldots, b\}$ and $y \leq x$, $W_{\mathbf{M}_c(T[m]),m}(y) = W_{\mathbf{M}_c(T[m])}(y)$.

Then x_m, a_m exist and $x_m = x$ and $a_m = a$. In particular, k_m is an index for $W_{\mathbf{M}_a(T[m])}$ which is L. This index is only computed from x_m, a_m which are for all sufficiently large m the same, thus both $\mathbf{M}_a$ and $\mathbf{M}_{b+1}$ converge on T to an index for L. ∎

Note that this proof covers the case where $b = 1$, although one can solve this case much easier by taking $\mathbf{M}_2 = \mathbf{M}_1$. Furthermore, for $b = 2$ the following noninclusion is witnessed.

Corollary 18 $[2,3]\mathbf{NUShTxtEx} \not\subseteq \mathbf{NUShTxtEx}$.

4 Vacillatory Learning with 2 Indices

As every $\mathbf{NUShTxtFex}_b$-learner can be simulated by a **NUShTxtEx**-learner identifying the same class, we have that $\mathbf{TxtEx} = \mathbf{NUShTxtFex}_b \subset \mathbf{TxtFex}_2$ for all $b > 1$. But, the next, quite surprising result shows that in the case of $\mathbf{TxtFex}_2$ one can avoid U-shaped learning behaviour if one gives up the constraint that the learner has to vacillate between *finitely* many indices. That is, $\mathbf{TxtFex}_2 \subseteq \mathbf{NUShTxtBc}$. In Theorem 20 it is shown that there is a uniform learner **U** which takes as input a set F of up to 2 indices and **NUShTxtBc**-identifies every $\{W_e : e \in F\}$. Additionally every hypothesis of **U** is a subset of a W_e, with $e \in F$. Then this result is combined with Theorem 16 to show the inclusion $\mathbf{TxtFex}_2 \subseteq \mathbf{NUShTxtBc}$. But before turning to Theorem 20, the following auxiliary proposition, which is useful in simplifying the proof of Theorem 20.

Next proposition says that one can uniformly find for any set $F = \{i', j'\}$ of indices two new indices i, j such that (a) the enumerations of W_i, W_j are different at every stage s, except when $W_{i,s} = W_{j,s} = \emptyset$ and (b) $W_i = W_{i'}$ and $W_j = W_{j'}$ whenever possible: the only exception is the case when $W_{i'}$ and $W_{j'}$ are equal, finite and nonempty — in which case one of the new sets W_i, W_j is equal to them and the other is a proper subset. The four conditions in the proposition ensure the above.

Proposition 19 *Given a set $F = \{i', j'\}$ one can compute a set $G(F) = \{i, j\}$ such that*

- *For all s, either $W_{i,s} \cup W_{j,s} = \emptyset$ or $W_{i,s} \neq W_{j,s}$;*
- *$W_i \subseteq W_{i'}$ and if $W_{i'}$ is infinite then $W_i = W_{i'}$;*
- *$W_j \subseteq W_{j'}$ and if $W_{j'}$ is infinite then $W_j = W_{j'}$;*
- *$\{W_{i'}, W_{j'}\} \subseteq \{W_i, W_j\}$.*

This also holds with $j' = i'$ for the case that $F = \{i'\}$.

Proof. We let $W_i = \bigcup_s W_{i,s}$ and $W_j = \bigcup_s W_{j,s}$ for the approximations obtained by the following inductive construction: $W_{i,0} = \emptyset$ and $W_{j,0} = \emptyset$. From stage s to $s+1$, the approximations are updated as follows:

$$W_{i,s+1} = \begin{cases} W_{i',s+1} & \text{if } W_{i',s+1} \neq W_{j',s+1}; \\ W_{i',s} & \text{if } W_{i',s} \neq W_{i',s+1} = W_{j',s+1}; \\ W_{i',s} & \text{if } W_{i',s} = W_{i',s+1} = W_{j',s+1} \neq W_{j',s}; \\ W_{i,s} & \text{if } W_{i',s} = W_{i',s+1} = W_{j',s} = W_{j',s+1}; \end{cases}$$

$$W_{j,s+1} = \begin{cases} W_{j',s+1} & \text{if } W_{i',s+1} \neq W_{j',s+1}; \\ W_{j',s+1} & \text{if } W_{i',s} \neq W_{i',s+1} = W_{j',s+1}; \\ W_{j',s} & \text{if } W_{i',s} = W_{i',s+1} = W_{j',s+1} \neq W_{j',s}; \\ W_{j,s} & \text{if } W_{i',s} = W_{i',s+1} = W_{j',s} = W_{j',s+1}. \end{cases}$$

We now verify the properties claimed. If $W_{i,s} \neq W_{j,s}$ or $W_{i',s+1} \neq W_{j',s+1}$ then $W_{i,s+1} \neq W_{j,s+1}$. If at stage $s+1$ either $W_{i'}$ or $W_{j'}$ receives new elements then $W_{i,s+1} \neq W_{j,s+1}$, $W_{i,s+1} \in \{W_{i',s}, W_{i',s+1}\}$ and $W_{j,s+1} \in \{W_{j',s}, W_{j',s+1}\}$. By induction, $W_{i,s+1} \neq W_{j,s+1}$ whenever at least one of them is different from $\emptyset$. Furthermore, $W_i = W_{i'}$ and $W_j = W_{j'}$ whenever one of these sets is infinite or these two sets are different. Only if $W_{i'} = W_{j'}$ and both sets are finite, one of the sets W_i, W_j will be a proper subset of its counterpart — which cannot be avoided due to the goal that $W_{i,s+1} \neq W_{j,s+1}$ whenever at least one of these sets is not empty. ∎

Theorem 20 *There is a learner* **U** *such that for every r.e. sets* L, H *and every set* F *of indices for* L, H *with* $|F| \leq 2$*,* **U NUShTxtBc***-identifies* $\{L, H\}$ *using the additional information* F*. Furthermore, for every* $\sigma \in seg(\mathbb{N})$*,*

(1) $W_{\mathbf{U}(F,\sigma)} \subseteq L$ *or* $W_{\mathbf{U}(F,\sigma)} \subseteq H$*;*
(2) *if* $L = H$ *and* L *is infinite then* $W_{\mathbf{U}(F,\sigma)} \in \{\emptyset, L\}$*.*

Note that $L = H$ *is explicitly permitted.*

Proof. Given F, let $G(F)$ be as in Proposition 19. $G(F)$ is used instead of F in the algorithm **U** below, so that at every relevant stage, $W_{i,s} \boldsymbol{\Delta} W_{j,s}$ is nonempty and thus the minimum of the difference of these sets exists — this then simplifies the algorithm and its analysis. For a finite string σ, $\mathbf{U}(F, \sigma)$ outputs an index k for a set W_k enumerated by the algorithm given in Figure 1.

Intuitively, given $G(F) = \{i, j\}$ and a text T, the aim of the algorithm is to simulate $r \in \{i, j\}$ which best matches content(T). The case of input being $\emptyset$ or $W_i = W_j$ is easy to handle. For other cases, on input σ, the algorithm aims to find the least element x in $W_i \boldsymbol{\Delta} W_j$ and then simulate W_r for that $r \in \{i, j\}$ which satisfies [$x \in W_r$ iff $x \in$ content(σ)]. This would be sufficient for learning in **TxtBc** model. However, we need to make sure that algorithm is non U-shaped. The main problem for ensuring non U-shapeness arises from the fact that the estimated value of x may not be correct (and thus one may simulate a correct grammar followed by simulating an incorrect grammar). To handle this problem, the algorithm tries to keep track of whether the estimated value

Uniform non U-shaped Behaviourally Correct Learner U
Parameter: F. Input: σ. Output: k, specified implicitly.
Algorithm to enumerate $W_k = \bigcup_r W_{k,r}$.
(Start) Let $u = |\sigma|$, $C = \text{content}(\sigma)$ and $s = 0$.
Let $W_{k,t} = \emptyset$ for all $t < |\sigma|$.
If $C = \emptyset$ or $W_{e,u} = \emptyset$ for all $e \in G(F)$, Then go to (Empty).
Let $\tau = \sigma[|\sigma| - 1]$.
Select i, j, x such that
(a) $\{i, j\} = G(F)$;
(b) $x = \min(W_{i,u} \mathbf{\Delta} W_{j,u})$;
(c) $x \in W_{i,u} \Leftrightarrow x \in C$.
Go to (Branch).
(Branch) If $C \cup W_{i,s} \subseteq W_{\mathbf{U}(F,\tau),u}$ and $y \in (W_{i,u} - W_{i,|\sigma|}) \cup (W_{j,u} - W_{j,|\sigma|})$ for some $y \leq x$ Then go to (Copy) Else go to (Enum).
(Enum) Let t be the maximal element of $\{s, \ldots, u\}$ such that one of the following conditions holds:
(Min) $t = s$;
(Equal) $C \subseteq W_{i,t} \cup W_{j,t} \subset W_{i,u} \cap W_{j,u}$;
(Inf) $C \subseteq W_{i,t} \subset W_{i,u} \wedge \forall y \leq x\, (y \notin (W_{i,u} - W_{i,|\sigma|}) \cup (W_{j,u} - W_{j,|\sigma|}))$;
(Diff) $C \subseteq W_{i,t} \subset W_{i,u}$ and $W_{j,u} = W_{i,s}$;
(Sub) $C \subseteq W_{i,t} \subseteq W_{\mathbf{U}(F,\tau),u}$;
(Exact) $C = W_{i,t}$ and $t = |\sigma|$.
Let $W_{k,u} = W_{i,t}$, update $s = t$, $u = u + 1$ and go to (Branch).
(Copy) Let $W_{k,u} = W_{\mathbf{U}(F,\tau),u}$, update $u = u + 1$ and go to (Copy).
(Empty) Let $W_{k,u} = \emptyset$, update $u = u + 1$ and go to (Empty).

Fig. 1. Algorithm to enumerate $W_{\mathbf{U}(F,\sigma)}$ from Theorem 20.

of x is indeed correct. If not, then the algorithm tries to spoil the simulation by either following the previous hypothesis (using the step (Copy)), or doing the current simulation only partially (decision whether to follow the previous hypothesis or do a (partial) simulation is done in the (Branch) instruction of the algorithm). However, this does not always work as when the input is finite, one may have already simulated the whole language by the time one discovers the error in x. To handle this, in the step (Enum), we do a *slowed down* simulation, which also depends on the interplay between the languages enumerated by the grammars i, j. Details of this is somewhat complex and depends on an extensive case analysis. The substep (Equal) in the algorithm aims to handle the case when $W_i = W_j$, substep (Inf) tries to handle the case when the input language in infinite, substep (Exact) handles the case when the input is finite and all of the input has already been seen. Substeps (Sub) and (Diff) are needed for taking care of some special cases (such as when one of the languages L, H is subset of the other). We now proceed formally.

Note the following: if the length of σ is 0 then its content is $\emptyset$ and the algorithm goes to (Empty). Thus $|\sigma| > 0$ when defining $\tau = \sigma[|\sigma| - 1]$. Furthermore, the goal of the condition (Min) in (Enum) is just to ensure that $W_{k,u} \subseteq W_{k,u+1}$.

We first show that the algorithm **TxtBc**-learns $\{\emptyset, L, H\}$. The learning of $\emptyset$ is clear since the algorithm goes to the label (Empty) and conjectures $\emptyset$ whenever the content of the input is empty set.

The role of L, H is symmetric, so assume that $L \neq \emptyset$ and T being a text for L. For input $T[n]$, let k_n, C_n, i_n, j_n, x_n be the corresponding parameters of the variables in the algorithm where i_n, j_n, x_n are only defined if the algorithm does not go to (Empty). One of the following two Cases 1.1, 1.2 applies.

(Case 1.1) $L = H$ and L is infinite.
Let n be the first step such that the algorithm does not go to the label (Empty). Note that $n > 0$; $T[n-1]$ exists and $W_{\mathbf{U}(F,T[n-1])} = \emptyset$. Thus the algorithm does not reach the label (Copy) and the indices i_n, j_n satisfy $W_{i_n} = W_{j_n} = L$. For every t and all sufficiently large u the condition (Equal) in (Enum) is satisfied and thus $W_{i_n,t} \subseteq W_{k_n}$. In particular $W_{i_n} = W_{k_n}$. For $m = n, n+1, \ldots$ one can see that either $W_{k_{m+1}} = W_{k_m}$ since the algorithm goes infinitely often through (Copy) or $W_{k_{m+1}} = W_{i_m} = L$ since the algorithm goes infinitely often through (Enum) and the condition (Equal) is eventually satisfied for every t. In particular, **U TxtBc**-learns L from T.

(Case 1.2) $L \neq H$ or L is finite.
The two sets L', H' enumerated by the indices in $G(F)$ satisfy $L' \neq H'$ and $L \in \{L', H'\}$. Let $x = \min(L' \boldsymbol{\Delta} H')$. Assume that n is so large such that content$(T[n]) \neq \emptyset$ and all $e \in G(F)$ and $y \in \mathbb{N}$ satisfy the following:

- if L is finite and $y \in L$ then $y \in C_n$ and $W_{e,n}(y) = W_e(y)$;
- if $y \leq x$ then $C_n(y) = L(y)$ and $W_{e,n}(y) = W_e(y)$.

This implies that $W_{e,n} \neq \emptyset$ for some $e \in G(F)$ and the algorithm does not go to label (Empty). Thus x_n, i_n, j_n are defined and i_n is the unique index in $G(F)$ with $W_{i_n} = L$. Since $W_{i_n,n}, W_{j_n,n}$ coincide with W_{i_n}, W_{j_n} up to x_n, the algorithm will never reach the label (Copy) and only enumerate elements of W_{i_n} into L, thus $W_{k_n} \subseteq W_{i_n}$. If L is finite then $L = C_n$ and the elements of L go into W_{k_n} due to the condition (Exact) in (Enum). If L is infinite then the algorithm will eventually enumerate all elements in any set $W_{i_n,t}$ since for every t there is a u with $W_{i_n,t} \subset W_{i_n,u}$ by condition (Inf). So in both cases, **U TxtBc**-learns L.

Additional property (2) has already been shown in Case 1.1. Now additional property (1) is verified inductively. For $n = 0$ the hypothesis is $\emptyset$ and the property is true. Having the property for n, consider the hypothesis $W_{k_{n+1}}$. If the algorithm for $W_{k_{n+1}}$ goes to label (Empty) then $W_{k_{n+1}} = \emptyset$ and the property holds for $n+1$. If the algorithm for $W_{k_{n+1}}$ reaches the label (Copy) then $W_{k_{n+1}} = W_{k_n}$ and the property is true by induction. Otherwise $W_{k_{n+1}} \subseteq W_{i_{n+1}}$ and $W_{i_{n+1}}$ is a subset of either L or H, so additional property (1) holds also in this case.

Now it is shown that the learner is non U-shaped. Let n be the first index with $W_{k_n} = L$. Note that $L = W_{k_n} \subseteq W_{i_n}$ (since (Copy) step was not used in W_{k_n}). Thus one of the following Cases 2.1, 2.2, 2.3, 2.4 applies.

Below let P_n denote the property:

> $W_{i_n} \cap \{0, \ldots, x_n\} = W_{i_n,n} \cap \{0, \ldots, x_n\}$ and $W_{j_n} \cap \{0, \ldots, x_n\} = W_{j_n,n} \cap \{0, \ldots, x_n\}$.

(Case 2.1) $L = W_{i_n}$ and $L \subset W_{\mathbf{U}(F,T[n-1])}$.

In this case clearly, $W_{i_n} \neq W_{j_n}$ via the additional property (1) proved above. Note that the algorithm on input $T[n]$ never reaches the label (Copy) in this case, since otherwise $W_{k_n} = W_{\mathbf{U}(F,T[n-1])} \neq L$. We now claim that P_n holds. To see this, note that if $L = W_{i_n}$ is finite, then P_n holds (otherwise the procedure would eventually go to (Copy) via (Branch)). If $L = W_{i_n}$ is infinite, then, since $W_{i_n} \neq W_{j_n}$, (Equal), (Diff), (Exact) cannot act infinitely often. Furthermore if (Inf) acts infinitely often, then clearly P_n holds. If (Sub) acts infinitely often, then P_n holds (otherwise the procedure would eventually go to (Copy) via (Branch)).

Furthermore $W_{\mathbf{U}(F,T[n-1])} \subseteq W_{j_n}$ and thus $x_n \in W_{j_n} - W_{i_n}$. Thus, for all $m \geq n$, $x_m = x_n$, $x_n \in W_{j_m,m} - W_{i_m,m}$ and $i_m = i_n$.

Now one shows by induction that $W_{i_m} = L$ for all $m \geq n$. It clearly holds for $m = n$. Assume now as inductive hypothesis for $m \geq n$ that $W_{k_m} = L$. If the algorithm for input $T[m+1]$ reaches (Copy) then $W_{k_{m+1}} = W_{k_m}$. If it goes through (Enum) infinitely often then on one hand $W_{k_{m+1}} \subseteq W_{i_{m+1}} = L$ and on the other hand $W_{k_{m+1}} \supseteq W_{i_{m+1},t}$ for all t with $C_m \subseteq W_{i_m,t}$ since for sufficiently large u the condition (Sub) is satisfied. Again $W_{k_{m+1}} = L$.

(Case 2.2) $L = W_{i_n}$ and $L \not\subseteq W_{\mathbf{U}(F,T[n-1])}$ and L is finite.

There is a first u where $W_{k_n,u} = L$. This cannot happen in (Copy), so the algorithm on input $T[n]$ goes to (Enum) infinitely often. In (Enum), W_{k_n} can become equal to W_{i_n} only due to the condition (Exact) since the conditions (Equal), (Inf), (Diff) give only proper subsets of L and (Sub) does not apply by $L \not\subseteq W_{\mathbf{U}(F,T[n-1])}$. Thus $C_n = L = W_{i_n,n} = W_{i_n}$. Thus, for all $m \geq n$, $C_m = L = W_{i_n,m} = W_{i_n}$, and hence $i_m = i_n$. Then one can show by induction as in the last paragraph of Case 2.1 that $W_{i_m} = L$ for all $m \geq n$.

(Case 2.3) $L = W_{i_n}$ and $L \not\subseteq W_{\mathbf{U}(F,T[n-1])}$ and L is infinite.

If $W_{i_n} = W_{j_n}$ then the condition (Sub) in (Enum) and (Copy) guarantee that $W_{i_m+1} = W_{i_m}$ for all $m \geq n$. If $W_{i_n} \neq W_{j_n}$ then W_{k_n} being infinite means that for every t with $C_n \subseteq W_{i,t}$ there is an $u \geq t$ such that one of the conditions (Equal), (Inf), (Diff), (Sub) and (Exact) are satisfied. Due to $W_{i_n} \neq W_{j_n}$, $W_{i_n} \not\subseteq W_{\mathbf{U}(F,T[n-1])}$ the conditions (Equal), (Diff) and (Sub) are satisfied only for finitely many t. Similarly (Exact) is satisfied only for $t = n$ if at all. Thus for almost all t, $W_{i,t}$ is enumerated into $W_{k_n,u}$ by satisfying condition (Inf). Thus all $y \in (W_{i_n} - W_{i_n,n}) \cup (W_{j_n} - W_{j_n,n})$ are strictly larger than x_n. So $x_m = x_n$ for all $m \geq n$. Since $W_{i_n} = L$, it follows that $x_n \in L$ iff $x_n \in W_{i_n}$ iff $x_n \in W_{i_n,n}$ iff $x_n \in \text{content}(T[n])$. Thus $i_m = i_n$ for all $m \geq n$. Now one can prove inductively for all $m \geq n$ that $W_{k_m} = L$ as in the last paragraph of Case 2.1.

(Case 2.4) $L \subset W_{i_n}$.

Then L is finite since $L = W_{i_n,s}$ for some s. Furthermore $L = W_{j_n}$ since

$L \neq W_{i_n}$. Since $W_{k_{n-1}} \neq W_{k_n}$, the algorithm on input $T[n]$ never goes to the label (Copy) and thus goes through (Enum) infinitely often. If u is so large that $W_{i_n,s} \subset W_{i_n,u}$ and $W_{j_n,u} = L$ then there cannot be any t with $W_{i_n,s} \subset W_{i_n,t} \subset W_{i_n,u}$ since otherwise the set W_{k_n} would be increased according to the condition (Diff) in (Enum). It follows that for all $t \geq s$, either $W_{i_n,t} = L$ or $W_{i_n,t} = W_{i_n}$. Now one can show by induction that $W_{k_m} = L$ for all $m \geq n$. Having this for W_{k_m}, consider the following cases for $W_{k_{m+1}}$: if the algorithm reaches label (Copy) then $W_{k_{m+1}} = W_{k_m}$. If it ends up going through (Enum) infinitely often and $i_{m+1} = j_n$ then $W_{k_{m+1}} = W_{j_n} = L$ due to condition (Sub) in (Enum). If it ends up going through (Enum) infinitely often and $i_{m+1} = i_n$ then $W_{k_{m+1}} \supseteq W_{i_n,s}$ by the condition (Sub). But no condition permits to get $W_{k_{m+1}} \supseteq W_{i_{m+1},t}$ for some t with $W_{i_{m+1},t} \supset W_{i_{m+1},s}$ since for this t there is no u with $W_{i_{m+1},t} \subset W_{i_{m+1},u}$ as required in the conditions (Equal), (Inf) and (Diff). Furthermore, (Exact) and (Sub) do not enforce $W_{i_{m+1},t} \subseteq W_{k_{m+1}}$ since $W_{i_{m+1},t}$ is neither a subset of C_{m+1} nor of W_{k_m}. So again $W_{k_{m+1}} = L$.

This completes the proof. ∎

Theorem 21 *Every* **TxtFex**$_2$*-learnable class is* **NUShTxtBc**-*learnable.*

Proof. Given a **TxtFex**$_2$-learnable class $\mathcal{L}$, there is by Theorem 6 a pair of two learners $\mathbf{N}_1, \mathbf{N}_2$ which converge on every language from $\mathcal{L}$ and $[1,2]$**TxtEx**-identify $\mathcal{L}$. Obtain from these two learners the team $\mathbf{M}_1, \mathbf{M}_2$ as done in the proof of Theorem 16. Let F be as defined in the proof of Theorem 16. Let σ_τ, D_τ be the values of σ, D computed on input τ by the algorithm for $\mathbf{M}_1, \mathbf{M}_2$ in the proof of Theorem 16. Note that one can, for each input σ_τ, D_τ, check in the limit whether $W_{F(\sigma_\tau,D_\tau,a)}$ enumerates some elements using part (f) of the algorithm in the proof of Theorem 16. Let $\mathbf{U}$ be as in Theorem 20.

Now one builds the following new learner $\mathbf{U}'$:

$$W_{\mathbf{U}'(\tau)} = W_{\mathbf{U}(\{e_1,e_2\},\tau)} \text{ where, for } a = 1,2,$$

$$W_{e_a} = \begin{cases} W_{F(\sigma_\tau,D_\tau,a)} & \text{if (f) is not used for } W_{F(\sigma_\tau,D_\tau,1)} \text{ or } W_{F(\sigma_\tau,D_\tau,2)}; \\ W_{F(\sigma_\tau,D_\tau,1)} \cup W_{F(\sigma_\tau,D_\tau,2)} & \text{if (f) is used for } W_{F(\sigma_\tau,D_\tau,1)} \text{ or } W_{F(\sigma_\tau,D_\tau,2)}. \end{cases}$$

Let $L \in \mathcal{L}$ and T be a text for L. Let n be the first number where one of the sets $W_{\mathbf{M}_1(T[n])}, W_{\mathbf{M}_2(T[n])}$ is L. Then, for any $m \geq n$ and any $a \in \{1,2\}$, $W_{\mathbf{M}_a(T[m])} = W_{\mathbf{M}_a(T[n])}$ and $W_{\mathbf{M}_a(T[m])}$ is not of the form $E \cup \tilde{E}$, where E is as introduced in the proof of Theorem 16 and $\tilde{E}$ is finite. Then $\mathbf{U}$ is fed with the same parameter set $\{e_1, e_2\}$ for all $m \geq n$ and one of the e_1, e_2 enumerates L. Thus $\mathbf{U}'$ **TxtBc**-learns L on T.

It remains to show that $\mathbf{U}'$ is non U-shaped on T. This is clearly true if L is the empty set. So assume $L \neq \emptyset$. Consider any m with $W_{\mathbf{U}'(T[m])} = L$. If case

(f) of the algorithm for $F(\sigma_{T[m]}, D_{T[m]}, 1)$ or $F(\sigma_{T[m]}, D_{T[m]}, 2)$ applies, then W_{e_1} and W_{e_2} are the same infinite set $E \cup \tilde{E}$ for some finite set $\tilde{E}$. It follows by the additional property (2) of $\mathbf{U}$ in Theorem 20 that $\mathbf{U}'(T[m])$ either outputs an index for the empty set or for $E \cup \tilde{E}$; both sets are different from L, thus case (f) does not apply. Hence, $T[m]$ is a stabilizing sequence for both $\mathbf{M}_1, \mathbf{M}_2$ on those sets $W_{\mathbf{M}_1(T[m])}, W_{\mathbf{M}_2(T[m])}$ which are not empty. Since one of these is a superset of L by the additional property (1) of $\mathbf{U}$ in Theorem 20, it follows that $\mathbf{M}_1, \mathbf{M}_2$ do not change mind on T beyond $T[m]$. For $a = 1, 2$ the parameter e_a is defined as above for $\tau = T[m]$ and it holds that $W_{e_a} = W_{\mathbf{M}_a(T[m])}$. Thus $\mathbf{U}'(T[o])$ coincides with $\mathbf{U}(\{e_1, e_2\}, T[o])$ for all $o \geq m$ and $\mathbf{U}$ with the parameter set $\{e_1, e_2\}$ is non U-shaped on the text T for L. The same holds for $\mathbf{U}'$. Thus $\mathbf{U}'$ **NUShTxtBc**-learns $\mathcal{L}$. ∎

5 Vacillatory Learning with 3 Indices

From Theorem 11 it is already known that U-shaped learning behaviour is necessary for $\mathbf{TxtFex}_b$ $(b > 1)$ identification of any class in $\mathbf{TxtFex}_b - \mathbf{TxtFex}_1$ for all $b > 1$. Theorem 22 strengthens this result by showing that, for some classes of languages in $\mathbf{TxtFex}_b$ for $b > 2$, the necessity of U-shaped behaviour cannot be circumvented by allowing infinitely many correct grammars in the limit, that is, by shifting to the more liberal criterion of **TxtBc**-identification. This is one of the rare cases in inductive inference where the containment in a class defined without numerical parameters holds for level 2 but not for level 3 and above of a hierarchy. The proof is a diagonalization proof reminiscent of the proof of Theorem 4 in [20].

Theorem 22 $\mathbf{TxtFex}_3 \not\subseteq \mathbf{NUShTxtBc}$.

Proof. Let $L_{i,j} = \{\langle i, j, k\rangle : k \in \mathbb{N}\}$, $I_{i,j} = W_i \cap L_{i,j}$ and $J_{i,j} = W_j \cap L_{i,j}$ for $i, j \in \mathbb{N}$. The class

$$\mathcal{L} = \{L_{i,j} : i, j \in \mathbb{N}\} \cup \{I_{i,j}, J_{i,j} : i, j \in \mathbb{N} \wedge I_{i,j} \subset J_{i,j} \wedge |I_{i,j}| < \infty\}$$

witnesses the separation.

To see that $\mathcal{L}$ is in $\mathbf{TxtFex}_3$, consider the following machines $\mathbf{N}_I, \mathbf{N}_J, \mathbf{N}_L$ which initially output indices of the empty set. Each of them waits for the first tuple of the form $\langle i, j, k\rangle$ for some k to come up in the input. From then on, $\mathbf{N}_I$ outputs an index for $I_{i,j}$ forever, $\mathbf{N}_J$ an index for $J_{i,j}$ forever and $\mathbf{N}_L$ an index for $L_{i,j}$ forever. So, for every $i, j \in \mathbb{N}$, $\mathbf{N}_I$ learns the set $I_{i,j}$, $\mathbf{N}_J$ the set $J_{i,j}$ and $\mathbf{N}_L$ the set $L_{i,j}$. The class $\mathcal{L}$ is learnable by a team of three machines which converge on every text for every language in $\mathcal{L}$ to some index. It follows from Theorem 6 that $\mathcal{L}$ is in $\mathbf{TxtFex}_3$.

So it remains to show that $\mathcal{L}$ is not in **NUShTxtBc**, that is, to show that any given **TxtBc**-learner for $\mathcal{L}$ is U-shaped on some text for some language in $\mathcal{L}$.

Given the learner $\mathbf{M}$, one defines the following function F by an approximation from below.

$$F_s(i,j) = \begin{cases} F_{s-1}(i,j) & \text{if } s > 0 \text{ and} \\ & W_{\mathbf{M}(\langle i,j,0\rangle\,\langle i,j,1\rangle\,\ldots\,\langle i,j,F_{s-1}(i,j)\rangle),s} \subseteq L_{i,j}; \\ k & \text{otherwise where } k \text{ is the first number} \\ & \text{found with } k > F_t(i+j)+s, \text{ for all } t < s, \text{ and} \\ & \{\langle i,j,0\rangle, \langle i,j,1\rangle, \ldots, \langle i,j,k\rangle\} \subset \\ & \qquad\qquad W_{\mathbf{M}(\langle i,j,0\rangle\,\langle i,j,1\rangle\,\ldots\,\langle i,j,k\rangle)}. \end{cases}$$

Since $\langle i,j,0\rangle, \langle i,j,1\rangle, \ldots$ is a text for $L_{i,j}$ and $\mathbf{M}$ **TxtBc**-learns $L_{i,j}$, almost all hypotheses $\mathbf{M}(\langle i,j,0\rangle\,\langle i,j,1\rangle\,\ldots\,\langle i,j,k\rangle)$ are indices for $L_{i,j}$. Thus the k is always found in the second part of the definition of F_s and F_s is well-defined. Furthermore, if $F_{s-1}(i,j)$ is sufficiently large, the condition $W_{\mathbf{M}(\langle i,j,0\rangle\,\langle i,j,1\rangle\,\ldots\,\langle i,j,F_{s-1}(i,j)\rangle),s} \subseteq L_{i,j}$ holds for all s and thus $F_s(i,j) = F_{s-1}(i,j)$. So the limit $F(i,j)$ of all $F_s(i,j)$ exists and is approximated from below. By considering the first s where $F(i,j) = F_s(i,j)$ and the fact that it is then no longer updated, one has

$$\{\langle i,j,0\rangle, \langle i,j,1\rangle, \ldots, \langle i,j,F(i,j)\rangle\} \subset W_{\mathbf{M}(\langle i,j,0\rangle\,\langle i,j,1\rangle\,\ldots\,\langle i,j,F(i,j)\rangle)} \subseteq L_{i,j}.$$

Now there are r.e. sets W_a, W_b such that

$$W_a = \{\langle i,j,l\rangle : i,j \in \mathbb{N} \wedge l \in \{0,1,\ldots,F(i,j)\}\},$$
$$W_b = \{\langle i,j,l\rangle : i,j \in \mathbb{N} \wedge (\exists t > l)\,[\langle i,j,l\rangle \in W_{\mathbf{M}(\langle i,j,0\rangle\,\langle i,j,1\rangle\,\ldots\,\langle i,j,F_t(i,j)\rangle),t}]\}.$$

Now fix the parameters i,j such that $i = a$ and $j = b$; the cases where $i \neq a$ or $j \neq b$ are not important in the considerations below.

Assume that $\langle i,j,l\rangle \in W_b$ using a parameter t with $F_t(i,j) \neq F(i,j)$. Let s be the first stage with $F_s(i,j) = F(i,j)$; note that $s > t$. Then by the definition of F_s, $F(i,j) = F_s(i,j) > s > t$ and $\{\langle i,j,0\rangle, \langle i,j,1\rangle, \ldots, \langle i,j,F_s(i,j)\rangle\} \subset W_{\mathbf{M}(\langle i,j,0\rangle\,\langle i,j,1\rangle\,\ldots\,\langle i,j,F_s(i,j)\rangle)}$. So $\langle i,j,l\rangle$ is in $W_{\mathbf{M}(\langle i,j,0\rangle\,\langle i,j,1\rangle\,\ldots\,\langle i,j,F(i,j)\rangle)}$ as well.

Thus $\{\langle i,j,0\rangle, \langle i,j,1\rangle, \ldots, \langle i,j,F(i,j)\rangle\} = W_i \cap L_{i,j} = I_{i,j} \subset J_{i,j} = W_j \cap L_{i,j} = W_{\mathbf{M}(\langle i,j,0\rangle\,\langle i,j,1\rangle\,\ldots\,\langle i,j,F(i,j)\rangle)}$ and $I_{i,j}$ is finite. Hence $I_{i,j}, J_{i,j} \in \mathcal{L}$.

Now consider a text T for $J_{i,j}$ formed as follows. Let σ be the sequence $\langle i,j,0\rangle\,\langle i,j,1\rangle\,\ldots\,\langle i,j,F(i,j)\rangle$. Note that $\mathbf{M}(\sigma)$ outputs an index for $J_{i,j}$. Let $\tau = \sigma\#^r$, for some r, be such that $\mathbf{M}(\tau)$ is an index for $I_{i,j}$. Note that there exists such τ since $\mathbf{M}$ **TxtBc**-identifies $I_{i,j}$. Let T be a text for $J_{i,j}$ starting with τ. Now $\mathbf{M}$ on T has to output an index $J_{i,j}$ beyond τ. Hence, $\mathbf{M}$ is U-shaped on text T, and thus $\mathbf{M}$ is not a **NUShTxtBc**-learner for $\mathcal{L}$. Since $\mathbf{M}$ was chosen arbitrarily, $\mathcal{L}$ is not **NUShTxtBc**-learnable. ∎

Intriguingly, the proof of Theorem 22 above features the contrast between learning a finite table and learning a general rule. This contrast is often invoked in accounts of U-shaped learning behaviour in children [9]. In our proof the finite table is embodied by the set $I_{i,j}$, while the (possibly) infinite set, only specified / learnable by a general rule is the sets $J_{i,j}, L_{i,j}$. The presence of both these types of sets is the key for the impossibility of learning (even in the **TxtBc** sense) the class $\mathcal{L}$ with a non U-shaped learner. Observe that the proof does *not* feature the learning of an incorrect general rule followed by a correct general rule augmented by a finite table, as in most psychological accounts of U-shaped learning behaviour. Actually the proof shows that any learner of $\mathcal{L}$ when confronted with the task of learning $J_{i,j}$ is forced to overgeneralize when fed the finite table $I_{i,j} = \{\langle i,j,0\rangle, \ldots, \langle i,j,F(i,j)\rangle\} \subset J_{i,j}$ which it also has to learn – $\mathbf{M}$ conjectures $J_{i,j}$ although the data is from the proper subset $I_{i,j}$. After that and seeing long enough only elements from $I_{i,j}$, $\mathbf{M}$ correctly learns this finite table. But then, it eventually returns to the general rule representing the set $J_{i,j}$ when more examples from this set come up.

Since $\mathbf{TxtFex}_3 \subset \mathbf{TxtFex}_4 \subset \ldots \subset \mathbf{TxtFex}_*$, one immediately gets the following corollary.

Corollary 23 $(\forall b \in \{3, 4, \ldots, *\})\, [\mathbf{TxtFex}_b \not\subseteq \mathbf{NUShTxtBc}]$.

A further corollary is that the counterpart of Theorem 20 does not hold for sets of three indices. Indeed, if such an algorithm would exist, then one could **NUShTxtBc**-learn $\mathcal{L}$ from Theorem 22 by conjecturing $\emptyset$ until the first triple $\langle i,j,k\rangle$ comes up and then simulating the uniform learner with indices for the sets $I_{i,j}, J_{i,j}, L_{i,j}$. However, by Theorem 22 such a learner does not exist.

Corollary 24 *No machine uniformly* **NUShTxtBc**-*learns* $\{W_e : e \in F\}$ *with* F *as additional information where* F *is a set of* 3 *indices.*

6 Teams Revisited

Classes in $\mathbf{TxtFex}_2$ are in **TxtBc** and in $[1, 2]$**NUShTxtEx**. The next proposition shows that one cannot weaken the condition of being in $\mathbf{TxtFex}_2$ to the combination of the two consequences in Theorem 21. Furthermore the condition that the team members converge on every text for a language in $\mathcal{L}$ is essential in Theorem 6.

Proposition 25 *The class* $\mathcal{L}$ *from Theorem 22 is* $[1, 2]$**NUShTxtEx**-*learnable and* $\mathbf{TxtFex}_3$-*learnable but it is not* **NUShTxtBc**-*learnable.*

Proof. By Theorem 22, $\mathcal{L}$ is $\mathbf{TxtFex}_3$-learnable but not **NUShTxtBc**-learnable. So it remains to show that $\mathcal{L}$ is $[1, 2]$**NUShTxtEx**-learnable.

Let L be a language and T be a text for L. The behaviour of the team on T is explained as follows where $C_n = \text{content}(T[n])$. As long as $C_n = \emptyset$, both learners output a fixed index for $\emptyset$. If $C_n \neq \emptyset$ then determine the components i, j of the first data item $\langle i,j,k\rangle$ in $T[n]$ which is not #. The first learner outputs a fixed index for $L_{i,j}$. The second learner considers an index e of $I_{i,j}$ computed from i, j. On input $T[n]$, the learner computes the least m satisfying the following:

- $W_{e,m} = W_{e,n}$;
- $C_m \neq \emptyset$;
- $C_m \subseteq W_{e,m} \Leftrightarrow C_n \subseteq W_{e,n}$.

Then the second learner outputs an hypothesis e_m which is given as

$$W_{e_m} = \begin{cases} L_{i,j} \cup L_{i+1,j+1} & \text{if } W_{e,m} \subset W_e; \\ I_{i,j} & \text{if } C_m \subseteq W_{e,m} \text{ and } W_{e,m} = W_e; \\ J_{i,j} & \text{if } C_m \not\subseteq W_{e,m} \text{ and } W_{e,m} = W_e. \end{cases}$$

Note that this hypothesis depends only on m and C_m but not on n and C_n. Now it is shown that the team correctly $[1,2]$**NUShTxtEx**-learns L from T. There are five cases, in the last four cases it is assumed that $L \in \mathcal{L}$.

$L \notin \mathcal{L}$: Then one only has to verify that the algorithm for both learners is recursive which can be done easily. By the way, this is also satisfied for all cases below.

$L = \emptyset$: Then it is easy to see that both members of the team always output the same index for $\emptyset$.

$L = I_{i,j}$ and $L \neq \emptyset$: Then $I_{i,j} \subset J_{i,j}$ and $I_{i,j}$ is finite. The first learner never outputs an hypothesis for $I_{i,j}$ and is thus non U-shaped. The second learner outputs incorrect hypotheses which are either $\emptyset$ or $L_{i,j} \cup L_{i+1,j+1}$ until n is so large that $W_{e,n} = I_{i,j}$ and $C_n \neq \emptyset$. From then on the second learner outputs the index e_m for the first m such that $C_m \neq \emptyset$ and $W_{e,m} = W_{e,n}$; this index e_m indeed enumerates $I_{i,j}$ and is never replaced by another one.

$L = J_{i,j}$ and $L \subset L_{i,j}$: Then $I_{i,j} \subset J_{i,j}$ and $I_{i,j}$ is finite. The first learner never outputs an hypothesis for $J_{i,j}$ and is thus non U-shaped. The second learner outputs incorrect hypotheses which are either $\emptyset$ or $L_{i,j} \cup L_{i+1,j+1}$ or $I_{i,j}$ until n is so large that $W_{e,n} = I_{i,j}$ and $C_n \not\subseteq I_{i,j}$. From then on the second learner outputs the index e_m for the first m such that $C_m \not\subseteq W_{e,n}$ and $W_{e,m} = W_{e,n}$; this index e_m indeed enumerates $J_{i,j}$ and is never replaced by another one.

$L = L_{i,j}$: Then the first learner outputs finitely often an index for $\emptyset$ and then makes exactly one mind change to an index for $L_{i,j}$. If $I_{i,j}$ is infinite then the second learner always outputs an hypothesis for $L_{i,j} \cup L_{i+1,j+1}$ which is incorrect. If $I_{i,j}$ is finite then the second learner outputs finitely often incorrect hypotheses until it makes a final mind change to an hypothesis for $J_{i,j}$, which can be verified as in the previous case. Thus the team is non U-shaped in both subcases, the one where $J_{i,j} = L_{i,j}$ and the one where $J_{i,j} \neq L_{i,j}$.

So the team witnesses that $\mathcal{L}$ is indeed $[1,2]$**NUShTxtEx**-learnable. ∎

A corollary of the result is that $\mathbf{TxtFex}_2 \subset [1,2]$**NUShTxtEx**. This can also be shown in general for all $b > 1$.

Proposition 26 $\mathbf{TxtFex}_b \subset [1,b]$**NUShTxtEx** *for all* $b \in \{2,3,\ldots\}$.

Proof. The inclusion is from Theorem 16. Its properness is witnessed by the class $\mathcal{L}$ of all sets $\mathbb{N} - F$ where $1 \leq |F| \leq b$. The learner $\mathbf{M}_a$ determines for input σ the least a numbers $x_1, \ldots, x_a \notin \text{content}(\sigma)$ and conjectures the set $\mathbb{N} - \{x_1, \ldots, x_a\}$. If $|F| = a$ then $\mathbf{M}_a$ **NUShTxtEx**-learns $\mathbb{N} - F$. If $|F| \neq a$ then $\mathbf{M}_a$ never outputs an hypothesis for $\mathbb{N} - F$. Thus, the team $\mathbf{M}_1, \ldots, \mathbf{M}_b$ is $[1, b]$**NUShTxtEx**-learning the class $\mathcal{L}$. It is well-known that $\mathcal{L}$ is not in **TxtBc** and thus also not in $\mathbf{TxtFex}_b$ whenever $b \geq 2$. ▮

Remark 27 $\mathbf{TxtFex}_* \not\subseteq [1, b]\mathbf{TxtEx}$ *for all* $b \in \mathbb{N}^+$*. This is witnessed by the class* $\{W_e : e \in \mathbb{N} \wedge \{0, 1, \ldots, e\} \subseteq W_e \subset \mathbb{N}\}$*.*

A further interesting question is whether one can at least obtain non U-shaped team learning for arbitrary team learnable classes. This is true for $[1, 1]$**TxtEx** by Theorem 16 but it fails for $[1, 2]$**TxtEx**-learning. We will show in Theorem 28 below that $[1, b]\mathbf{NUShTxtEx} \subset [1, b]\mathbf{TxtEx}$. Before proving the theorem, we give intuitive idea of the construction for $b = 2$ case.

Baliga, Case, Merkle, Stephan and Wiehagen [4] constructed an infinite class $\mathcal{C}$ which satisfies: (a) $\mathcal{C} \in \mathbf{TxtEx}$, (b) $\mathbb{N} \notin \mathcal{C}$, (c) for every finite set S, all but finitely many languages in $\mathcal{C}$ are supersets of S and (d) $\mathcal{C}$ cannot be non U-shaped **TxtEx**-learnt by a learner which does not output an index for $\mathbb{N}$. Property (d) holds even if one considers only the class $\{L \in \mathcal{C} : S \subseteq L\}$, for any fixed finite set S. (Note that [4] actually used decisive instead of non U-shaped learning. However their argument essentially works for non U-shaped learning too).

Now, $\mathcal{C} \cup \{\mathbb{N}\}$ witnesses $[1, 2]\mathbf{NUShTxtEx} \subset [1, 2]\mathbf{TxtEx}$. To see this note that using (a) above, one easily gets $\mathcal{C} \cup \{\mathbb{N}\} \in [1, 2]\mathbf{TxtEx}$. For the diagonalization against non U-shaped learners $\mathbf{M}_1, \mathbf{M}_2$, one first notes that if any of these machines outputs a grammar for $\mathbb{N}$ on σ, then it outputs a grammar for $\mathbb{N}$ on any extension of σ (otherwise non U-shaped property is violated). Now to identify $\mathbb{N}$, one of the machines (say $\mathbf{M}_1$) must output a grammar for $\mathbb{N}$ on some input σ. Thus, $\mathbf{M}_1$ cannot learn any language in $\mathcal{C}$ which extends content(σ). If $\mathbf{M}_2$ outputs a grammar for $\mathbb{N}$ on some extension τ of σ, then the team $\mathbf{M}_1, \mathbf{M}_2$ cannot learn any extension of content(τ) except $\mathbb{N}$. On the other hand if $\mathbf{M}_2$ does not output a grammar for $\mathbb{N}$ on any extension of σ, then it essentially learns (in non U-shaped manner) $\{L \in \mathcal{C} : \text{content}(\sigma) \subseteq L\}$, without outputting a grammar for $\mathbb{N}$ — an impossible task by (d) above.

The proof of theorem below uses the above idea by essentially using $\{L : \text{card}(\mathbb{N} - L) \leq b - 2\}$ instead of just $\{\mathbb{N}\}$ (along with a modification of $\mathcal{C}$ mentioned above). The technical details become somewhat complicated due to the interaction of above method with the technique of [4].

Theorem 28 *For all* $b \in \{2, 3, \ldots\}$*,* $[1, b]\mathbf{NUShTxtEx} \subset [1, b]\mathbf{TxtEx}$.

Proof. Let $b \in \{2, 3, \ldots\}$. Since the inclusion is obvious from the definition, one only has to show that it is a proper inclusion. Consider an r.e. list $team_0, team_1, \ldots$ of all teams of b **TxtEx**-learners. Let $D_0, D_1, \ldots$ be an enumeration of all sets of $b - 1$ elements such that $m < n$ whenever

$\max(D_m) < \max(D_n)$. We say that $team_m$ *qualifies at* n iff there is an ordering $\mathbf{M}_1, \ldots, \mathbf{M}_b$ of the machines in $team_m$ and a string $\sigma_{m,n} \in \text{seg}(\mathbb{N})$ such that, for z_k denoting the k-th nonelement of content$(\sigma_{m,n})$, following four conditions hold.

(1) $m \leq n$ and $|\sigma_{m,n}| \geq 2z_{b-1}$ and $D_n = \{z_1, \ldots, z_{b-1}\}$;
(2) $\sigma_{m,n}[2z_{b-1}] \in \text{seg}(\{0, \ldots, z_{b-1}\})$;
(3) for $a = 1, \ldots, b-1$ and $L_a = \mathbb{N} - \{z_c : 1 \leq c < a\}$, $\sigma_{m,n}[2z_{b-1}]$ is a stabilizing sequence for $\mathbf{M}_a$ on L_a;
(4) $\text{content}(\sigma_{m,n}) \cup \{z_b, z_{b+1}, \ldots, z_{2b}\} \subseteq W_{\mathbf{M}_b(\sigma_{m,n})}$.

One can determine with oracle K whether $team_m$ qualifies at n — first one can test for all orderings of the b team members of $team_m$ and for all $\tau \in \text{seg}(\{0, \ldots, z_{b-1}\})$ of length $2z_{b-1}$ whether they are stabilizing sequences for $\mathbf{M}_a$ on L_a for $a = 1, \ldots, b-1$. For such τ and orderings of the teams, one can then check whether there is an extension $\sigma_{m,n}$ which also satisfies the other conditions. If $team_m$ does not qualify at n then let $G_{m,n} = \{0, \ldots, \max(D_n)\} - D_n$. Otherwise fix the first string $\sigma_{m,n}$ found, let $H_{m,n} = W_{\mathbf{M}_b(\sigma_{m,n})}$ and let $G_{m,n}$ be the set $\text{content}(\sigma_{m,n}) \cup \{z_k\}$ for the first $k \geq b$ where

$$\text{content}(\sigma_{m,n}) \cup \{z_k\} \notin \{W_{\mathbf{M}_1(\sigma_{m,n})}, \ldots, W_{\mathbf{M}_b(\sigma_{m,n})}\}.$$

Note that there are only b machines and thus $k \leq 2b$. So $G_{m,n} \subset W_{\mathbf{M}_b(\sigma_{m,n})}$.

The class $\mathcal{L}$ to be learned is constructed using an inductively defined sequence $u_0, u_1, \ldots$ of auxiliary numbers.

(1) Put all subsets of $\mathbb{N}$ into $\mathcal{L}$ which miss at most $b-2$ elements of $\mathbb{N}$;
(2) For every n, put $G_{0,n}, G_{1,n}, \ldots, G_{n,n}$ into $\mathcal{L}$;
(3) For each n search for the least m satisfying the following conditions.
  - $2^m \leq \min(D_n)$;
  - there is a value $h(m,n)$, with $m \leq h(m,n) \leq n$, such that $team_m$ qualifies at $h(m,n)$ and $(\forall x \leq \max(D_n))\, [x \in H_{m,h(m,n)} \Leftrightarrow x \notin D_n]$;
  - $u_{n'} \neq m$ for all $n' < n$.

  If such an m is found and hence $h(m,n)$ defined,
  Then let $u_n = m$ and put $H_{m,h(m,n)}$ into $\mathcal{L}$
  Else let $u_n = \infty$ and put $\mathbb{N} - D_n$ into $\mathcal{L}$.

The symbol ∞ is just used to have a value different from all natural numbers for $u_0, u_1, \ldots$ if needed.

To see that $\mathcal{L}$ is in $[1,b]\mathbf{TxtEx}$, consider the following learners $\mathbf{N}_1, \ldots, \mathbf{N}_b$. For $a = 1, \ldots, b-1$, the learner $\mathbf{N}_a$ always computes the set E of the least $a-1$ numbers which have not yet shown up in the input and outputs a canonical index for $\mathbb{N} - E$. It is easy to see that all sets which miss at most $b-2$ elements of $\mathbb{N}$ are learned by a member of this team. In particular, $\mathbf{N}_1$ always conjectures $\mathbb{N}$. The last learner $\mathbf{N}_b$ deals with all sets $L \in \mathcal{L}$ having at least $b-1$ nonelements. Let $u_{n,s}, G_{m,n,s}, h_s(m,n)$ be recursive approximations to $u_n, G_{m,n}$ and $h(m,n)$, respectively. Note that these exist as the construction above is recursive in K. The algorithm of $\mathbf{N}_b$ on input τ is the following:

(1) Let n be the number for which D_n is the set of the least $b-1$ nonelements of content(τ);
(2) If content$(\tau) = G_{m,n,|\tau|}$ for some $m \in \{0,\ldots,n\}$
Then output a canonical index for content(τ)
Else if $u_{n,|\tau|} = \infty$ then output a canonical index for $\mathbb{N} - D_n$
Else compute the $|\tau|$-th approximation for the index of $H_{u_{n,|\tau|},h_{|\tau|}(u_{n,|\tau|},n)}$.

If τ is long enough and $L = G_{m,n}$ then $\mathbf{N}_b$ clearly learns the set. If $L \neq G_{m,n}$ for the finitely many sets of this form in $\mathcal{L}$, then $\mathbf{N}_b$ will converge to an index of $H_{m,h(m,n)}$ (for the case that $u_n = m$) or to an index of $\mathbb{N} - D_n$ (for the case that $u_n = \infty$). So $\mathcal{L}$ is in $[1,b]$**TxtEx**.

It remains to show that $\mathcal{L}$ is not in $[1,b]$**NUShTxtEx**. Suppose by way of contradiction that $team_m$ $[1,b]$**NUShTxtEx**-infers $\mathcal{L}$. We claim that there is an n such that $team_m$ qualifies at n and $G_{m,n}, H_{m,n}$ are both in $\mathcal{L}$. So assume by way of contradiction that such an n does not exist. Then there is in particular no n with $u_n = m$ since otherwise $G_{m,h(m,n)}, H_{m,h(m,n)}$ both exist and are in $\mathcal{L}$. The following arguments are now used to get the desired contradiction.

Let $\tau_0 = 0\,1\,\ldots\,m$ and $z_0 = m$ in order to get that $z_1 > m$ below. Do the following for $a = 1,\ldots,b-1$:

(1) Choose a machine $\mathbf{M}_a$ from $team_m$ which has a locking sequence extending τ_{a-1} on $\mathbb{N} - \{z_k : 0 < k < a\}$;
(2) Choose z_a and an extension τ_a of τ_{a-1} satisfying the following additional constraints:
  - $z_1 < \cdots < z_a$ are the least a nonelements of content(τ_a);
  - content$(\tau_a) = \{0,\ldots,z_a - 1\} - \{z_k : 0 < k < a\}$ and $z_{a-1} < z_a$ and $2z_a = |\tau_a|$;
  - τ_a is a locking sequence for $\mathbf{M}_a$ on $\mathbb{N} - \{z_k : 0 < k < a\}$;
  - If $a = b-1$ then the n with $D_n = \{z_1,\ldots,z_{b-1}\}$ satisfies $u_n = \infty$ and $n > n'$ for all n' with $u_{n'} < m$.

Note that the fourth condition in (2) can be satisfied in the case that $a = b-1$ since there is, for each m', at most one n' with $u_{n'} = m'$ and furthermore there are less than $1+\log(z_1)$ many n with $\min(D_n) \leq z_1 \wedge u_n < \infty$. Since $u_n = \infty$, $\mathbb{N} - D_n$ is a member of $\mathcal{L}$. Let $\mathbf{M}_b$ be the unique machine in $team_m$ different from $\mathbf{M}_1,\ldots,\mathbf{M}_{b-1}$. Since $\mathbf{M}_b$ has to infer $\mathbb{N} - D_n$ on every text extending τ_{b-1} for that language, there is an extension τ_b of τ_{b-1} which is a locking sequence for $\mathbf{M}_b$ on $\mathbb{N} - D_n$.

It is easy to see that τ_b would be a possible choice for $\sigma_{m,n}$, thus $team_m$ qualifies at n and $G_{m,n}, H_{m,n}$ exist. $G_{m,n}$ is in $\mathcal{L}$. If $H_{m,n} \notin \mathcal{L}$ then $H_{m,n}$ must miss out at least $b-1$ elements of $\mathbb{N}$. Let n' be such that $D_{n'}$ is the set of the least $b-1$ nonelements of $H_{m,n}$. Since $G_{m,n} \subseteq H_{m,n}$ and D_n is the set of the least $b-1$ nonelements of $G_{m,n}$, either $D_n = D_{n'} \wedge n = n'$ or $\max(D_n) < \max(D_{n'}) \wedge n < n'$. But then $u_{n'} > m$ although m would qualify as a value of $u_{n'}$, since $H_{m,n}$ exists and n would be a possible value for

$h(m, n')$ — that is $h(m, n')$ would be defined and $G_{m,h(m,n')}, H_{m,h(m,n')} \in \mathcal{L}$ in contradiction to the assumption on the nonexistence of these sets in $\mathcal{L}$.

So for the $team_m$ there is an n such that $team_m$ qualifies at n and $H_{m,n}$ is in $\mathcal{L}$. Since $G_{m,n} \subseteq \mathbb{N} - D_n$, $\sigma_{m,n}$ is a stabilizing sequence for the first $b-1$ machines on $G_{m,n}$. Furthermore, these $b-1$ machines output a grammar for a languages different from $G_{m,n}$ on input $\sigma_{m,n}$. Finally $G_{m,n} \subset H_{m,n}$ and the last machine $\mathbf{M}_b$ conjectures $H_{m,n}$ on $\sigma_{m,n}$. Now let $\tau = \sigma_{m,n}\#^r$ be such that $\mathbf{M}_b$ on τ outputs a grammar for $G_{m,n}$ (such a τ exists, since otherwise $team_m$ does not $[1, b]$**TxtEx**-identify $G_{m,n}$). Let T be a text for $H_{m,n}$, which extends τ. Now, $\mathbf{M}_b$ gives up the correct hypothesis (output at $\sigma_{m,n}$) for input text T. This contradicts the assumption that $team_m$ $[1, b]$**NUShTxtEx**-learns $\mathcal{L}$ and completes the proof. ∎

Proposition 29 $[a, b]$**TxtEx** $\subseteq [a, a+b]$**NUShTxtEx**.

Proof. Angluin [1] defined that a learner is conservative iff every mind change from an hypothesis e to a new hypothesis e' is justified in the sense that some data already seen are not contained in W_e at the time of the mind change. So every conservative learner is non U-shaped since it never abandons a correct hypothesis. The new team will consist of conservative learners and thus be non U-shaped.

Given now a team $\mathbf{M}_1, \ldots, \mathbf{M}_b$ which $[a, b]$**TxtEx**-identifies some class $\mathcal{L}$, there is for every $c \in \{1, \ldots, b\}$ a machine $\mathbf{N}_c$ which is conservative and **TxtEx**-learns every infinite language **TxtEx**-learned by $\mathbf{M}_c$ [24,32]. So every infinite language in $\mathcal{L}$ is learned by at least a of the conservative machines $\mathbf{N}_1, \ldots, \mathbf{N}_b$. Furthermore, one assigns to the machines $\mathbf{N}_{b+1}, \ldots, \mathbf{N}_{b+a}$ the algorithm which outputs on every input σ a canonical index for content(σ). These machines are conservative as well and each of them learns every finite set. Thus the new team $\mathbf{N}_1, \ldots, \mathbf{N}_{a+b}$, $[a, a+b]$**NUShTxtEx**-learns $\mathcal{L}$. ∎

7 Summary and Final Discussion

The following results were obtained.

- $\mathbf{TxtFex}_b \subset [1, b]$**NUShTxtEx** for all $b \in \{2, 3, \ldots\}$.
- $\mathbf{TxtFex}_1 =$ **TxtEx** $=$ **NUShTxtEx** $= [1, 1]$**NUShTxtEx**, see also [4].
- $[1, b]$**NUShTxtEx** $\subset [1, b]$**TxtEx**, for all $b \in \{2, 3, \ldots\}$.
- $\mathbf{NUShTxtFex}_b =$ **NUShTxtEx** for all $b \in \{1, 2, \ldots, *\}$.
- $\mathbf{TxtFex}_2 \subseteq$ **NUShTxtBc**.
- $\mathbf{TxtFex}_3 \not\subseteq$ **NUShTxtBc**.

These results and the facts known from previous work [4,11] are summarized in Figure 2. Single-headed arrows in the diagram denote proper inclusions. Double-headed arrows denote equality. All transitive closures of the inclusions

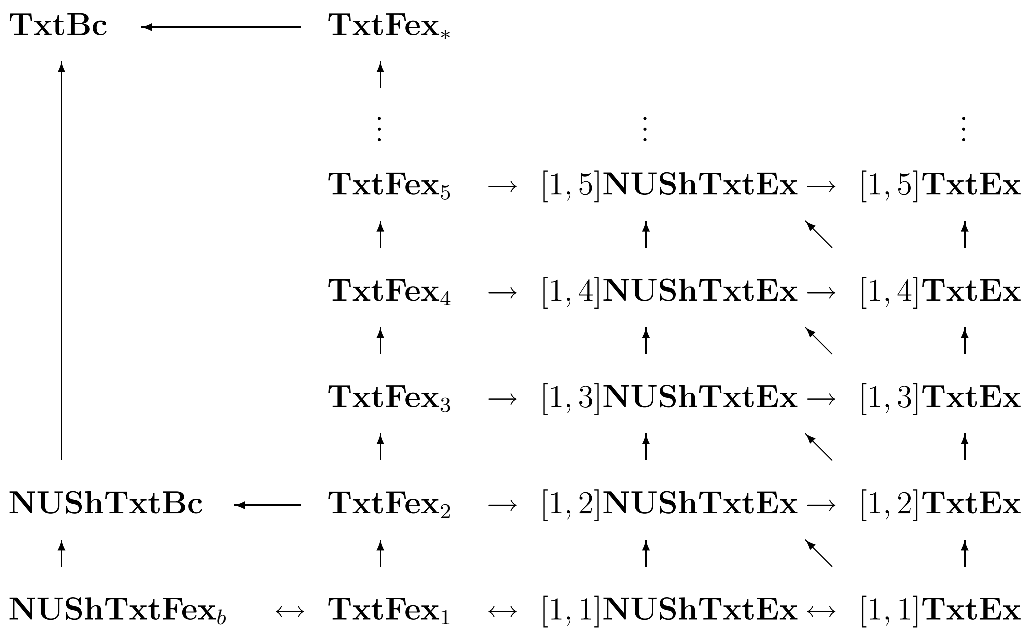

Fig. 2. Summary of the results for $b \in \{1, 2, 3, 4, 5, *\}$.

displayed are valid and no other inclusions hold between language learning criteria in the diagram.

Some of the results obtained in this paper and [4] are suggestive for what may be true for the case of human cognition. For example, perhaps the class of tasks humans must learn to be competitive in the genetic marketplace *necessitates* U-shaped learning behaviour, like the classes featured in our Theorem 22.

We note that our proof that $\mathbf{TxtFex}_3 \not\subseteq \mathbf{NUShTxtBc}$ intriguingly features, as already observed, learning finite tables versus general rules, as in the common account of the U-shaped behaviour by sustainers of the symbolic theory (see for example, [9]). Nevertheless, our proof does *not*, as might be expected from those accounts, feature, among other things, learning an incorrect general rule *followed* by learning a general rule augmented by a correct finite table. The *order* of the interplay between a general rule and a finite table is different in our proofs. This difference may be significant or, more likely, nothing more than an artifact of our particular proof.

Not explored herein, but very interesting to investigate in the future, is the possible necessity of U-shaped learning in complexity-bounded learning contexts such as those explored in [12,13,16,40,47].

Note that our notion of non U-shaped team learning does not follow the standard philosophy of team learning where on every text of a given language, a out of the b machines learn this language (with some constraints on behaviour such as non U-shapedness) and there is no restriction on the behaviour of the remaining $b-a$ machines. An anonymous referee proposed also to consider this notion of Modified non U-shaped $[a, b]\mathbf{TxtEx}$ but it unfortunately coincides with $[a, b]\mathbf{TxtEx}$. This can be seen as follows. Given $\mathcal{L} \in [a, b]\mathbf{TxtEx}$, one can construct a team of b learners such that for every $L \in \mathcal{L}$, a out of the b team members **TxtEx**-identify L from every text: let $\mathcal{L}_i$ be the class of all languages identified by the i-th learner in this team. Now by Theorem 9, $\mathcal{L}_i$ is in **NUShTxtEx**, as witnessed by some machine $\mathbf{M}_i$. The machines $\mathbf{M}_1, \ldots, \mathbf{M}_b$ thus witness that $\mathcal{L}$ is learnable under the modified criterion for non U-shaped $[a, b]$**TxtEx**. Hence, we did not pursue this alternative further.

8 Acknowledgements

We thank an anonymous referee for many helpful comments.

References

[1] Dana Angluin. Inductive inference of formal languages from positive data. *Information and Control*, 45:117–135, 1980.

[2] Ganesh Baliga, John Case and Sanjay Jain. Language learning with some negative information. *Journal of Computer and System Sciences*, 51:273–285, 1995.

[3] Ganesh Baliga, John Case, Wolfgang Merkle and Frank Stephan. Unlearning helps. In U. Montanari, J. Rolim and E. Welzl, editors, *Proceedings of the 27th International Colloquium on Automata, Languages and Programming*, Lecture Notes in Computer Science 1853, pages 844–855. Springer-Verlag, 2000.

[4] Ganesh Baliga, John Case, Wolfgang Merkle, Frank Stephan and Rolf Wiehagen. Unlearning helps. Long version of [3], in preparation.

[5] Janis Bārzdiņš. Two theorems on the limiting synthesis of functions. In *Theory of Algorithms and Programs, vol. 1*, pages 82–88. Latvian State University, 1974. In Russian.

[6] Lenore Blum and Manuel Blum. Toward a mathematical theory of inductive inference. *Information and Control*, 28:125–155, 1975.

[7] Manuel Blum. A machine independent theory of the complexity of the recursive functions. *Journal of the Association of Computing Machinery*, 14:322–336, 1967.

[8] T. G. R. Bower. Concepts of development. In *Proceedings of the 21st International Congress of Psychology*. Presses Universitaires de France, pages 79–97, 1978.

[9] Melissa Bowerman. Starting to talk worse: Clues to language acquisition from children's late speech errors. In S. Strauss and R. Stavy, editors, *U-Shaped Behavioral Growth*. Academic Press, New York, 1982.

[10] S. Caray. An analysis of a learning paradigm. In S. Strauss and R. Stavy, editors, *U-Shaped Behavioral Growth*, Developmental Psychology Series. Academic Press, 1982.

[11] John Case. The power of vacillation in language learning. *SIAM Journal on Computing*, 28(6):1941–1969, 1999.

[12] John Case, Sanjay Jain, Steffen Lange and Thomas Zeugmann. Incremental concept learning for bounded data mining. *Information and Computation*, 152:74–110, 1999.

[13] John Case, Sanjay Jain and Arun Sharma. Complexity issues for vacillatory function identification. *Information and Computation*, 116(2):174–192, February 1995.

[14] John Case and Chris Lynes. Machine inductive inference and language identification. In M. Nielsen and E. M. Schmidt, editors, *Proceedings of the 9th International Colloquium on Automata, Languages and Programming*, Lecture Notes in Computer Science 140, pages 107–115. Springer-Verlag, 1982.

[15] John Case and Carl H. Smith. Comparison of identification criteria for machine inductive inference. *Theoretical Computer Science*, 25:193–220, 1983.

[16] C. C. Florencio. Consistent identification in the limit of some Penn and Buszkowski's classes is NP-hard. In *Proceedings of the International Conference on Computational Linguistics*, 1999.

[17] Rusins Freivalds, Efim Kinber and Rolf Wiehagen. On the power of inductive inference from good examples. *Theoretical Computer Science*, 110:131–144, 1993.

[18] Rūsiņš Freivalds and Carl H. Smith On the Role of procrastination for machine learning. *Information and Computation*, 107:237–271, 1993.

[19] Mark Fulk. Prudence and other conditions on formal language learning. *Information and Computation*, 85:1–11, 1990.

[20] Mark Fulk, Sanjay Jain and Daniel Osherson. Open problems in "Systems That Learn". *Journal of Computer and System Sciences*, 49:589–604, 1994.

[21] William Gasarch and Carl H. Smith Learning via queries. *Journal of the Association of Computing Machinery* 39(3):649–676, 1992.

[22] E. Mark Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[23] Sanjay Jain, Daniel Osherson, James Royer and Arun Sharma. *Systems that Learn: An Introduction to Learning Theory.* MIT Press, 1999. Second Edition of Reference [32].

[24] Sanjay Jain and Arun Sharma. Generalization and specialization strategies for learning r.e. languages. *Annals of Mathematics and Artificial Intelligence*, 23:1–26, 1998.

[25] Efim Kinber and Frank Stephan. Language learning from texts: mind changes, limited memory and monotonicity. *Information and Computation*, 123:224–241, 1995.

[26] David Kirsh. PDP learnability and innate knowledge of language. In S. Davis, editor, *Connectionism: Theory and Practice*, pages 297–322. Oxford University Press, 1992.

[27] Steffen Lange and Thomas Zeugmann. Monotonic versus non-monotonic language learning. In *Proceedings of the Second International Workshop on Nonmonotonic and Inductive Logic*, Lecture Notes in Artificial Intelligence 659, pages 256–269. Springer-Verlag, 1993.

[28] Gary Marcus, Steven Pinker, Michael Ullman, Michelle Hollander, T. John Rosen and Fei Xu. *Overregularization in Language Acquisition*. Monographs of the Society for Research in Child Development, vol. 57, no. 4. University of Chicago Press, 1992. Includes commentary by Harold Clahsen.

[29] David McNeill. Developmental psycholinguistics. In F. Smith and G. A. Miller, editors, *The Genesis of Language*, pages 15–84. MIT Press, 1966.

[30] Piergiorgio Odifreddi. *Classical Recursion Theory.* North Holland, Amsterdam, 1989.

[31] Daniel Osherson and Scott Weinstein. A note on formal learning theory. *Cognition*, 11:77–88, 1982.

[32] Daniel Osherson, Michael Stob and Scott Weinstein. *Systems that Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists*. MIT Press, 1986.

[33] Daniel Osherson and Scott Weinstein. Criteria of language learning. *Information and Control*, 52:123–138, 1982.

[34] Steven Pinker. Formal models of language learning. *Cognition*, 7:217–283, 1979.

[35] Kim Plunkett and Virginia Marchman. U-shaped learning and frequency effects in a multi-layered perceptron: implications for child language acquisition. *Cognition*, 38(1):43–102, 1991.

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

[37] Gisela Schäfer-Richter. Über Eingabeabhängigkeit und Komplexität von Inferenzstrategien. Ph.D. Thesis, Rheinisch-Westfälische Technische Hochschule Aachen, Germany, 1984.

[38] Carl H. Smith. The power of pluralism for automatic program synthesis. *Journal of the Association of Computing Machinery*, 29:1144–1165, 1982.

[39] Carl H. Smith. Three Decades of Team Learning. In Setsuo Arikawa and Klaus P. Jantke (editors), *Algorithmic Learning Theory, 4th International Workshop on Analogical and Inductive Inference, AII 1994, 5th International Workshop on Algorithmic Learning Theory, ALT 1994, Reinhardsbrunn Castle, Germany, October 10-15, 1994, Proceedings*, *Lecture Notes in Computer Science*, 872:211-228, 1994.

[40] Werner Stein. Consistent polynomial identification in the limit. In M. M. Richter, C. H. Smith, R. Wiehagen and T. Zeugmann, editors, *Algorithmic Learning Theory: Ninth International Conference (ALT' 98)*, volume 1501 of *Lecture Notes in Artificial Intelligence*, pages 424–438. Springer-Verlag, 1998.

[41] Sidney Strauss and Ruth Stavy, editors. *U-Shaped Behavioral Growth*. Developmental Psychology Series. Academic Press, 1982.

[42] Sidney Strauss, Ruth Stavy and N. Orpaz. The child's development of the concept of temperature, 1977. Manuscript, Tel-Aviv University.

[43] N.A. Taatgen and J.R. Anderson. Why do children learn to say broke? A model of learning the past tense without feedback. *Cognition*, 86(2):123–155, 2002.

[44] Kenneth Wexler. On extensional learnability. *Cognition*, 11:89–95, 1982.

[45] Kenneth Wexler and Peter W. Culicover. *Formal Principles of Language Acquisition*. MIT Press, 1980.

[46] Rolf Wiehagen. Characterization problems in the theory of inductive inference. In G. Ausiello and C. Böhm, editors, *Proceedings of the Fifth International Colloquium on Automata, Languages and Programming*, Lecture Notes in Computer Science 62, pages 494–508. Springer-Verlag, 1978.

[47] Rolf Wiehagen and Thomas Zeugmann. Ignoring data may be the only way to learn efficiently. *Journal of Experimental and Theoretical Artificial Intelligence*, 6:131–144, 1994.