

Iterative Learning from Texts and Counterexamples Using Additional Information

Sanjay Jain^{*1} and Efim Kinber²

¹ School of Computing, National University of Singapore, Singapore 117417, Republic of Singapore. Email: sanjay@comp.nus.edu.sg

² Department of Computer Science, Sacred Heart University, Fairfield, CT 06825-1000, U.S.A. Email: kinbere@sacredheart.edu

Abstract. A variant of *iterative* learning in the limit (cf. [LZ96]) is studied when a learner gets negative examples refuting conjectures containing data in excess of the target language and uses additional information of the following four types: a) memorizing up to n input elements seen so far; b) up to n *feedback* memberships queries (testing if an item is a member of the input seen so far); c) the number of input elements seen so far; d) the maximal element of the input seen so far. We explore how additional information available to such learners (defined and studied in [JK07]) may help. In particular, we show that adding the maximal element or the number of elements seen so far helps such learners to infer any *indexed* class of languages *class-preservingly* (using a descriptive numbering defining the class) — as it is proved in [JK07], this is not possible without using additional information. We also study how, in the given context, different types of additional information fare against each other, and establish hierarchies of learners memorizing $n + 1$ versus n input elements seen and $n + 1$ versus n feedback membership queries.

Keywords: Inductive Inference, Iterative Learning, Feedback, Memory Limitation.

1 Introduction

In this paper, we study some variants of learning in the limit from positive data and negative counterexamples to conjectures, with restricted access to input data. The general framework for study of learning in the limit was introduced in [Gol67]. In Gold’s original model, **TextEx**, a learner is able to hold full input data seen so far in its long-term memory. However, this assumption is apparently too strong for modeling many learning and cognitive processes. Wiehagen in [Wie76] (see also [LZ96]) suggested a model for learning in the limit where the long-term memory of the learners is limited to what they can store in their conjectures. These learners are called *iterative* learners. This learning model, while strongly limiting long-term memory, still makes salient an important aspect of learnability in the limit: its incremental character. Some variants of iterative learning proved to be quite useful in the context of applied machine learning (for

^{*} Supported in part by NUS grant numbers R252-000-308-112 and C-252-000-087-001.

example, [LZ06] applies the idea of iterative learning in the context of training Support Vector Machines).

The iterative learning model has been used for study of learnability from all positive examples (the corresponding formal model being denoted as **TxtIt**) as well as all positive and negative examples (denoted as **InfIt**, see [LZ92]). One can argue that **TxtIt** may be too weak (a learner gets only positive data and can memorize only very limited amount of input), whereas **InfIt** may be too strong: it is hard to conceive a realistic learning process, where the learner would be able to get access to full negative data. For example, children learning languages, while getting some negative data (in the form of corrections by parents or teachers), never get the full set of negative data.

In [JK08], the model **TxtEx** was extended to allow *negative counterexamples* to conjectures by a learner. This model is an example of *active learning*, where a learner communicates with a *teacher* (formally, an oracle) making *queries* and getting responses from the teacher. Active learning as a general framework for study of learning processes was introduced by D. Angluin in [Ang88] and has been widely utilized in various studies of theoretical and applied models of learnability from examples since then. The model of iterative learning from full positive data and negative counterexamples, **NCIt** (NC here stands for “*negative counterexample*”), defined in [JK07] actually combines two approaches: Gold’s framework (as the learner incrementally gets access to full positive data) and active learning (the learner, using *subset* queries, checks with the teacher if each conjecture does not contain data in excess of the target languages and if the answer is negative, the learner gets a *negative counterexample* showing an error). In linguistic terms, non-grammatical sentences in conjectures are, thus, being corrected. It must be noted that K. Popper [Pop68] regarded refutation of overgeneralizing conjectures as a vital part of learning and discovery processes.

In this paper, we extend the **NCIt** model to incorporate some additional features. Specifically, we consider the following two extensions of this model: in addition to subset queries (for conjectures), the learner

- a) can ask up to n *feedback* queries: whether the queried element belongs to the input seen so far;
- b) can store up to n input elements seen so far in its long-term memory (note that when the long-term memory used by a learner is n -bounded, if the memory is full then, in order to save a new input datum, the learner must sacrifice at least one element currently stored in the memory).

In the context of iterative learning of languages from positive data, these two types of “looking back” (in the context of feedback — using just one query per conjecture) were defined in [LZ96] (an earlier variant of memory-bounded learning can be found in [OSW86], and the idea of feedback learning goes back to [Wie76], where it was applied in the context of learning recursive functions in the limit). Both these concepts were reformalized (the former named *n-feedback* learning, and the latter named *n-memory* learning) and thoroughly studied and discussed in [CJLZ99]. Motivation for these sorts of learnability models, as discussed in [CJLZ99], comes from the rapidly developing field of knowledge discovery in databases, which includes, in particular,

data mining, knowledge extraction, information discovery, data pattern processing, information harvesting, etc. Many of these tasks represent interactive incremental iterative processes (cf., e.g., [BA96] and [FPSS96]), working on huge data sets, finding regularities, and verifying them on small samples of the overall data. While the authors in [CJLZ99] explore the aforementioned formalizations of “looking back” at small (uniformly limited by some upper bound n) portions of input data in the context of regular iterative learning, we, in this research, allow the learner to test with the teacher if conjectures do not contain data in excess of the target language. Our learners may also be allowed to memorize some bounds derived from the input data seen so far — in the form of the maximal element or the length of input seen so far (the latter type of additional information for iterative learners was first considered in [CM08]). In this research, we study how the aforementioned types of additional information can enhance capabilities of the **NCIt**-learnability model in general, and how they, while helping a learner, fare against each other.

Iterative learners from positive data, even with access to limited memory or feedback and/or additional information (in the form of, say, the maximal element or the number of elements seen so far) are obviously weaker than the learners being able to store full positive data seen so far. Our study shows how much extra learning power such iterative learners can obtain from a teacher in the form of a finite number of negative counterexamples to conjectures. As some of our results show, such negative counterexamples obtained from the teacher, even with extra help in the form of some additional information, are not able to compensate for even one extra memory cell or one feedback query (Theorems 15 and 16). Likewise, such negative examples do not help in the context of Theorems 17 and 18, which compare memory versus feedback. All these results significantly strengthen their counterparts in [CJLZ99]. Proving negative parts of these results required developing new technique utilizing limitations of extra learning power provided by a finite number of negative counterexamples; classes and proofs used for diagonalization reflect the advantages of extra feedback/memory, and are somewhat more transparent than the proofs used in [CJLZ99] for the feedback and memory hierarchy. On the other hand, there are situations in which *negative* counterexamples and some additional information (such as the maximal element or the number of elements in the input seen so far) give advantages to iterative learners over having bounded feedback (see, for example, details of the proof of Theorem 23; in a weaker context, advantages of testing conjectures with the teacher are demonstrated by Proposition 20: provided just some information about positive data seen so far, the iterative learners, at any moment, can test their conjectures with the teacher to retrieve positive data they are unable to store).

Specifically, in Section 3, we discover some general effects of additional information on **NCIt**-learners. In particular, it was established in [JK07] that iterative learners getting access to full positive and full negative data are, surprisingly, weaker than **NCIt**-learners (note that the latter ones get negative data just in the form of a finite set of negative counterexamples — however, only when these negative data are really necessary). We show that when the learners getting full positive and full negative data are allowed to memorize just one input datum or ask just one feedback membership query, they can sometimes learn more than any learner that gets

access to full positive data, that can use negative counterexamples (to conjectures), and that can store *all* data seen so far in its long-term memory (see Theorem 10). A known capability of **NCIt**-learners (established in [JK07]) is of special importance for many practical classes of languages: they can learn every indexed class of languages (that is, any class of recursive languages, where it is decidable, for any index k and any element m , if m is a member of the language with index k ; examples of such classes are the classes of all regular languages and all *pattern* languages ([Ang80])). However, as it was established in [JK07], **NCIt**-learners sometimes cannot learn an indexed class *class-preservingly* (cf. [LZZ08])— that is, they cannot learn by using any descriptive numbering defining just the target class as the hypothesis space. It turns out that this result regarding **NCIt**-learners holds even if the **NCIt**-learners are allowed to make n -feedback membership queries (see Theorem 14). However, class-preserving learning becomes possible if an **NCIt**-learner gets access to either the maximal element or to the number of elements seen so far (see Theorem 13), and thus by using 1-memory.

In Section 4, we strengthen some results in [CJLZ99], establishing non-trivial hierarchies of **NCIt**-learners using n -feedback queries or n -memory based on the number n (see Theorems 15 and 16). Our examples of classes witnessing the hierarchies in question also show that additional information in the form of the maximal element seen so far and the number of elements seen so far might not match the help that an **NCIt**-learner gets in the form of one extra feedback membership query, or one extra long-term memory cell.

In Section 5, we study tradeoffs between different types of additional information used by **NCIt**-learners (the main purpose of this study is to make salient advantages of each type of additional information for the learners in question). In particular, similarly to corresponding results in [CJLZ99], we show that one memory cell used by an **NCIt**-learner can give more help than any n feedback membership queries (even in presence of the maximal element and the number of elements seen so far), see Theorem 17, and, conversely, one feedback membership query can give more help than n -memory (plus the maximal element and the number of elements seen so far), see Theorem 18. Interestingly, the maximal element seen so far alone can give more help than any number of feedback membership queries, see Theorem 23. Also, the number of elements and the maximal element seen so far combined together can provide more help than any bounded number of memory cells or feedback membership queries, see Theorem 24. We also show how an extra memory cell can simulate the maximal element seen so far for **NCIt**-learners using n memory cells, see Proposition 19. We obtain also some partial results for other possible tradeoffs. Additionally, in Section 6 we demonstrate that yet another type of additional information, the length of input seen so far (note that this number may be, apparently, different from the number of elements in the input seen so far), in the context of **NCIt**-learning, can be replaced by the maximal element seen so far, see Theorem 31.

In Section 7, we briefly address the issue of robustness of our results using the maximal element and/or the number of elements seen so far in the presence of some “noise”. Namely, we discuss which results hold if the maximal element and the number of elements seen so far are replaced by upper bounds on these numbers, or by some approximations of these numbers.

2 Preliminaries

2.1 Notation

For any unexplained recursion theoretic notation we refer the reader to [Rog67]. The symbol \mathbb{N} denotes the set of natural numbers, $\{0, 1, 2, 3, \dots\}$. Languages are subsets of \mathbb{N} . Symbols \emptyset , \subseteq , \subset , \supseteq , and \supset , respectively, denote the empty set, subset, proper subset, superset, and proper superset. The cardinality of a set S is denoted by $\text{card}(S)$. The maximum and the minimum of a set are denoted by $\max(\cdot)$, $\min(\cdot)$, respectively, where $\max(\emptyset) = 0$ and $\min(\emptyset) = \infty$. The symbol \forall^∞ denotes ‘for all but finitely many’.

Let D_x denote the finite set with the canonical index x [Rog67]. We let $\langle \cdot, \cdot \rangle$ denote a pairing function, which is an arbitrary, computable, 1–1 mapping from $\mathbb{N} \times \mathbb{N}$ onto \mathbb{N} [Rog67]. We assume without loss of generality that $\langle \cdot, \cdot \rangle$ is monotonically increasing in both of its arguments. The corresponding projection functions are $\pi_1^2(\langle x, y \rangle) = x$ and $\pi_2^2(\langle x, y \rangle) = y$. The pairing function can be extended to n -tuples in a natural way (for example, by using $\langle x, y, z \rangle = \langle x, \langle y, z \rangle \rangle$). The corresponding projection functions are $\pi_i^n(x_1, x_2, \dots, x_n) = x_i$.

By φ we denote a fixed *acceptable* programming system for the partial computable functions from \mathbb{N} to \mathbb{N} [Rog67, HU79]. By φ_i we denote the partial computable function computed by the program with the number i in the φ -system. For a partial function η , $\eta(x) \downarrow$ denotes that $\eta(x)$ is defined. $\eta(x) \uparrow$ denotes that $\eta(x)$ is undefined.

By Φ we denote an arbitrary fixed Blum complexity measure [Blu67, HU79] for the φ -system. Intuitively, $\Phi_i(x)$ may be thought as the number of steps it takes to compute $\varphi_i(x)$.

By W_i we denote $\text{dom}(\varphi_i)$. Thus, W_i can be viewed as the recursively enumerable (r.e.) set/language accepted by the φ -program i . We also say that i is a grammar for W_i . By $W_{i,s}$ we denote the set $\{x < s : \Phi_i(x) < s\}$. The symbol \mathcal{E} denotes the set of all r.e. languages. We let L , with or without decorations³, range over \mathcal{E} . Symbol \mathcal{L} , with or without decorations, ranges over subsets of \mathcal{E} . The characteristic function of L is denoted by χ_L . We denote the complement of L by \bar{L} , that is, $\bar{L} = \mathbb{N} - L$.

We often need padding to code some arguments in a hypothesis. To this end, $\text{pad}(j, \cdot, \cdot, \dots)$ denotes a 1–1 recursive padding function (of an appropriate number of arguments) such that $W_{\text{pad}(j, \cdot, \cdot, \dots)} = W_j$ (such computable functions exist [Rog67]).

A class \mathcal{L} is said to be an *indexed class* iff there exists an indexing L_0, L_1, \dots of all and only the languages in \mathcal{L} such that for some recursive function f , $f(i, x) = \chi_{L_i}(x)$.

2.2 Basic Definitions for Learning

A *text* T is a mapping from \mathbb{N} into $(\mathbb{N} \cup \{\#\})$. $T(i)$ represents the $(i + 1)$ -th element in the text. We let T , with or without decorations, range over texts. The content of a text T , denoted, $\text{content}(T)$, is the set of natural numbers in the range of T . A text T is for a language L iff $\text{content}(T) = L$. Intuitively, $T(i)$ denotes the element presented to the learner at time i , and $\#$ ’s

³ Decorations are subscripts, superscripts, primes and the like.

represent pauses in the presentation of data. The initial sequence of a text T of the length n is denoted by $T[n]$, that is $T[n] = T(0)T(1) \dots T(n-1)$.

Sets of the form $\{x : x < n\}$, for some n , are called initial segments of \mathbb{N} . A (*finite*) *sequence* σ is a mapping from an initial segment of \mathbb{N} into $(\mathbb{N} \cup \{\#\})$. The empty sequence is denoted by λ . The set of all finite sequences is denoted by SEQ . We let σ, τ , and γ , with or without decorations, range over finite sequences. The *length* of σ , denoted by $|\sigma|$, is the number of elements in σ . For $n \leq |\sigma|$, $\sigma[n]$ denotes the initial sequence of σ of the length n . Thus, $\sigma[0]$ is λ . The content of σ , denoted $\text{content}(\sigma)$, is the set of natural numbers in the range of σ . We denote the sequence formed by the concatenation of τ at the end of σ by $\sigma \diamond \tau$. For simplicity of notation, sometimes we omit \diamond , when it is clear that concatenation is meant.

An *informant* [Gol67] I is a mapping from \mathbb{N} to $(\mathbb{N} \times \{0, 1\})$ such that for no $x \in \mathbb{N}$, both $(x, 0)$ and $(x, 1)$ are in the range of I . The content of I , denoted $\text{content}(I)$, is the set of pairs in the range of I . We say that I is an *informant* for L iff $\text{content}(I) = \{(x, \chi_L(x)) : x \in \mathbb{N}\}$. The *canonical informant* for L is the informant $(0, \chi_L(0))(1, \chi_L(1)) \dots$. Intuitively, informants give both all positive and all negative data for the language being learned. By $I[n]$ we denote the first n elements of the informant I .

An *inductive inference machine* (IIM) [Gol67] learning from texts is an algorithmic device which computes a (possibly partial) mapping from SEQ into \mathbb{N} . One can similarly define learners from informants and other modes of input as considered below. We use the term learner or learning machine as synonyms for inductive inference machines. In this paper we will only be considering learners which are algorithmic. We let M , with or without decorations, range over IIMs. We interpret $M(T[n])$ (or $M(I[n])$) as the grammar (index for an accepting program) conjectured by the IIM M on the initial sequence $T[n]$ (or $I[n]$). We say that M converges on T to i , (written: $M(T) \downarrow = i$) iff $M(T[n])$ is defined for all n and $(\forall n) [M(T[n]) = i]$. Convergence on informants is similarly defined.

There are several criteria for an IIM to be successful on a language. In this paper we will be mainly concerned with explanatory (abbreviated **Ex**) criteria of learning. Explanatory learning is also called learning in the limit.

Definition 1. [Gol67, CL82]

- (a) M **TextEx-identifies** an r.e. language L (written: $L \in \mathbf{TextEx}(M)$) iff for all texts T for L , there exists a grammar i such that $M(T) \downarrow = i$ and $W_i = L$.
- (b) M **TextEx-identifies** a class \mathcal{L} of r.e. languages (written: $\mathcal{L} \subseteq \mathbf{TextEx}(M)$) iff M **TextEx-identifies** each language from \mathcal{L} .
- (c) $\mathbf{TextEx} = \{\mathcal{L} \subseteq \mathcal{E} : (\exists M)[\mathcal{L} \subseteq \mathbf{TextEx}(M)]\}$.

As the learner only sees finitely many data before converging to its final hypothesis, some form of learning must have taken place. Thus, we use the terms identify, infer, and learn as synonyms.

Definition 2. [Gol67, CL82]

- (a) M **InfEx-identifies** an r.e. language L (written: $L \in \mathbf{InfEx}(M)$) iff for all informants I for L , there exists a grammar i such that $M(I) \downarrow = i$ and $W_i = L$.

- (b) M **InfEx**-identifies a class \mathcal{L} of r.e. languages (written: $\mathcal{L} \subseteq \mathbf{InfEx}(M)$) iff M **InfEx**-identifies each language from \mathcal{L} .
- (c) $\mathbf{InfEx} = \{\mathcal{L} \subseteq \mathcal{E} : (\exists M)[\mathcal{L} \subseteq \mathbf{InfEx}(M)]\}$.

Next we consider iterative learning.

Definition 3. [Wie76,LZ96]

- (a) M is iterative, iff there exists a partial recursive function F such that, for all T and n , $M(T[n+1]) = F(M(T[n]), T(n))$. Here $M(\lambda)$ is viewed as some predefined hypothesis.
- (b) M **TxtIt**-identifies \mathcal{L} , iff M is iterative, and M **TxtEx**-identifies \mathcal{L} .
- (c) $\mathbf{TxtIt} = \{\mathcal{L} : (\exists M)[M \text{ **TxtIt**-identifies } \mathcal{L}]\}$.

InfIt can be defined similarly. Note that for explanatory learning from informants, learning from canonical informants is enough to imply learnability from all informants. However, this is not the case for iterative learning. Thus, we have explicitly required learnability from all informants in the definitions above.

Intuitively, an iterative learner [Wie76,LZ96] is a learner whose every hypothesis depends only on its previous conjecture and current input. That is, for some recursive function F , for $n \geq 0$, $M(T[n+1]) = F(M(T[n]), T(n))$. Here, note that $M(T[0])$ is predefined to be some constant value. We will often identify F above with M (that is, use $M(p, x) = F(p, x)$ to describe $M(T[n+1])$, where $p = M(T[n])$ and $x = T(n)$). This is for ease of notation. Context determines which interpretation of the learner M is meant.

For **Ex** models of learning (for learning from texts or informants or their variants when learning from positive data and negative counterexamples, as defined below), one may assume without loss of generality that the learners are total, that is, defined on all initial segments of all texts (see, for example [OSW86]). However for iterative learning one cannot assume so. Thus, we explicitly require in the definition that iterative learners are defined on all inputs which are initial segments of texts (informants) for a language in the class.

Note that, although it is not stated explicitly, an **It**-type learner might store some input data in its conjecture (thus serving as a limited long-term memory). However, the amount of stored data cannot grow indefinitely, as the learner must converge to one (right) conjecture.

Using a standard coding of Turing machines, one can get a recursive enumeration M_0, M_1, \dots of all (iterative) IIMs which learn from texts, informants or negative counterexamples, based on context. Note that these IIMs might not be total.

Definition 4. (a) [Ful90] σ is said to be a **TxtEx**-stabilizing sequence for M on L , iff (i) $\text{content}(\sigma) \subseteq L$, and (ii) for all τ such that $\text{content}(\tau) \subseteq L$, $M(\sigma\tau) = M(\sigma)$.

(b) [BB75,Ful90] σ is said to be a **TxtEx**-locking sequence for M on L , iff (i) σ is a **TxtEx**-stabilizing sequence for M on L and (ii) $W_{M(\sigma)} = L$.

If M **TxtEx**-identifies L , then every **TxtEx**-stabilizing sequence for M on L is a **TxtEx**-locking sequence for M on L . Furthermore, one can show that if M **TxtEx**-identifies L , then for every σ such that $\text{content}(\sigma) \subseteq L$, there exists a **TxtEx**-locking sequence, which extends σ , for M on L

(see [BB75,Ful90]). Note that one can search (in the limit) for a stabilizing sequence for a learner M on a language L — though one may not be able to determine if it is a locking sequence (even in the limit).

Similar results can be shown for **InfEx**, **TxtIt**, **InfIt** and other criteria of learning discussed in this paper. We will often drop **TxtEx** (and other criteria notation) from **TxtEx**-stabilizing sequence and **TxtEx**-locking sequence, when the criterion is clear from context.

Learning with feedback and learning with bounded memory is a generalization of iterative learning where the learner has access to some past data using queries or via some finite amount of memory. Thus, in feedback learning an iterative learner is additionally allowed to query whether some elements were present in the past input data. In bounded memory, an iterative learner is able to memorize in its memory some (bounded) finite number of data (in addition to its latest conjecture). Below are the formal definitions.

Definition 5. [CJLZ99]

(a) Suppose M is a learning machine from texts (for a class \mathcal{L} of languages). We say that M is an m -feedback learner iff there exist partial recursive functions F and Q such that for all $L \in \mathcal{L}$, and all texts T for L ,

(i) for all n : $Q(M(T[n]), T(n)) \downarrow \in \mathbb{N}^m$, and

(ii) If $Q(M(T[n]), T(n)) = (x_1, x_2, \dots, x_m)$, then $M(T[n+1]) = F(M(T[n]), T(n), y_1, y_2, \dots, y_m)$, where $y_i = 1$ iff $x_i \in \text{content}(T[n+1])$.

(b) We say that M **TxtIt**-identifies \mathcal{L} with m -feedback iff M **TxtEx**-identifies \mathcal{L} and M is an m -feedback learner. Such learners M are also called **TxtIt**-learners using m -feedback.

Note that in the above definition, the feedback query on the input $T(n)$ is answered based on whether the queried element is included in the set $\text{content}(T[n+1])$. The results of this paper would not change if we considered answering the queries based on whether the queried element had been included in the set $\text{content}(T[n])$ (the proofs can be easily modified to handle this).

Definition 6. [LZ96]

(a) Suppose M is a learning machine from texts (for a class \mathcal{L} of languages). We say that M is an m -memory bounded learner iff there exists a partial recursive memory function **mem** (mapping finite sequences to finite sets) and partial recursive functions F, F' such that for all $L \in \mathcal{L}$, and all texts T for L ,

(i) for all n : $\mathbf{mem}(T[n]) \downarrow \subseteq \text{content}(T[n])$ and $\text{card}(\mathbf{mem}(T[n])) \leq m$

(ii) for all n : $\mathbf{mem}(T[n+1]) = F'(M(T[n]), T(n), \mathbf{mem}(T[n])) \downarrow$, and $\mathbf{mem}(T[n+1]) - \mathbf{mem}(T[n]) \subseteq \{T(n)\}$.

(iii) $M(T[n+1]) = F(M(T[n]), T(n), \mathbf{mem}(T[n])) \downarrow$.

(b) We say that M **TxtIt**-identifies \mathcal{L} with m -memory iff M **TxtEx**-identifies \mathcal{L} and M is an m -memory bounded learner. Such learners M are also called **TxtIt**-learner using m -memory or m -memory bounded **TxtIt**-learner.

In both the above definitions, $M(T[0])$ is some fixed initial hypothesis.

Again, we often identify the learner M with the function F (along with identifying **mem** with F') as defined above, and the context determines which interpretation of the learner M is meant.

One can similarly define feedback and memory bounded learning for learning from informants.

Besides the above models of learning, we sometimes allow the learner access to the maximal element in the input seen so far, or the number of elements in the input seen so far as an additional input, besides the input element $T(n)$ together with the latest conjecture and feedback/memory (and the counterexamples, in the case of **NC**-type learning defined below). In the sequel, we will typically refer to the “maximal element in the input seen so far” and the “number of elements in the input seen so far” as simply the “maximal element” and, respectively, the “number of elements”. Here note that, on an input text T , when $T(n)$ is presented to the learner, the maximal element given is $\max(\text{content}(T[n+1]))$ and the number of elements given is $\text{card}(\text{content}(T[n+1]))$ (and not $\max(\text{content}(T[n]))$ and $\text{card}(\text{content}(T[n]))$ respectively).

2.3 Learning with Negative Counterexamples

In this section we consider our models of learning from full positive data and negative counterexamples as given by [JK08]. Intuitively, for learning with negative counterexamples, we may consider the learner being provided a text, one element at a time, along with a negative counterexample to the latest conjecture, if any. (One may view this negative counterexample as a response of the teacher to the *subset query* when it is tested if the language generated by the conjecture is a subset of the target language). One may model the list of negative counterexamples as a second text for negative counterexamples being provided to the learner. Thus the IIMs get as input two texts T, T' , one for positive data, and the other for negative counterexamples.

We say that $M(T, T')$ converges to a grammar i (written: $M(T, T') \downarrow = i$) iff $M(T[n], T'[n])$ is defined for all n and $(\forall^\infty n)[M(T[n], T'[n]) = i]$.

First, we define the model of learning from positive data and negative counterexamples. In this model, if a conjecture contains elements not in the target language, then a negative counterexample is provided to the learner. **NC** in the definition below stands for *negative counterexample*.

Definition 7. [JK08]

(a) M **NCEx**-identifies a language L (written: $L \in \text{NCEx}(M)$) iff for all texts T for L , and for all T' satisfying the condition:

$$(T'(n) \in S_n, \text{ if } S_n \neq \emptyset) \text{ and } (T'(n) = \#, \text{ if } S_n = \emptyset), \\ \text{where } S_n = \overline{L} \cap W_{M(T[n], T'[n])}$$

$M(T, T')$ converges to a grammar i such that $W_i = L$.

(b) M **NCEx**-identifies a class \mathcal{L} of languages (written: $\mathcal{L} \subseteq \text{NCEx}(M)$), iff M **NCEx**-identifies each language in the class.

(c) $\text{NCEx} = \{\mathcal{L} : (\exists M)[\mathcal{L} \subseteq \text{NCEx}(M)]\}$.

Intuitively, S_n in the definition above is the set of negative counterexamples for the conjecture $W_{T[n], T'[n]}$ (any of which may be given to the learner). We call the least element of S_n the *least* (negative) counterexample (for the conjecture $W_{M(T[n], T'[n])}$). As the learner learning from negative counterexamples also must be able to learn from least negative counterexamples, in some cases, to simplify the diagonalization proofs we use least negative counterexamples.

For ease of notation, we sometimes define $M(T[n], T'[n])$ also as $M(T[n])$, where we separately describe how the counterexamples $T'(n)$ are presented to the conjecture of M on the input $T[n]$.

As an example, $\mathcal{L} = \{\mathbb{N}\} \cup \{L : \text{card}(L) < \infty\}$ is in **NCEx** but not in **TextEx**.

One can similarly define **NCIt**-learning, where the learner's output depends only on the previous conjecture, the latest positive data, and the counterexample provided.

Definition 8. [JK07]

- (a) M is iterative (for learning from positive data and negative counterexamples), iff there exists a partial recursive function F such that, for all T, T' and n , $M(T[n+1], T'[n+1]) = F(M(T[n], T'[n]), T(n), T'(n))$. Here $M(\lambda, \lambda)$ is some predefined constant.
- (b) M **NCIt**-identifies \mathcal{L} , iff M is iterative, and M **NCEx**-identifies \mathcal{L} .
- (c) **NCIt** = $\{\mathcal{L} : (\exists M)[M \text{ NCIt-identifies } \mathcal{L}]\}$.

We will often identify F above with M (that is, use $M(p, x, y) = F(p, x, y)$ to describe $M(T[n+1], T'[n+1])$, where $p = M(T[n], T'[n])$ and $x = T(n), y = T'(n)$). This is for ease of notation.

One should also note that the **NCIt** model is equivalent to allowing finitely many subset queries (with counterexamples for the answer “no”) in iterative learning.

One can extend the above definition to **NCIt**-learning with m -feedback or m -memory, by allowing the learner M up to m queries about whether some element x has appeared in the previous text or allowing the learner M to remember up to m elements of the past data. The resulting criteria are called **NCIt**-learning with m -feedback and **NCIt**-learning with m -memory, respectively. The resulting learners are called m -feedback **NCIt**-learner (or **NCIt**-learner using m -feedback) and m -memory bounded **NCIt**-learner (or **NCIt**-learner using m -memory), respectively.

It follows from the definition that **NCIt**-learning is contained in **NCIt**-learning using m -feedback and **NCIt**-learning using m -memory, which, in turn, are contained in **NCEx**.

To give the reader some intuition regarding how additional information can enhance the capabilities of **NCIt**-learners, consider the following example. Let

$\mathcal{L}_1 = \{L : W_{\min(L)} = L, \text{ and for all } x, \{2x, 2x+1\} \not\subseteq L\}$, and

$\mathcal{L}_2 = \{L : (\exists \text{ a unique } x)[\{2x, 2x+1\} \subseteq L] \text{ and } L = W_{\min(L)} \cup \{2x, 2x+1\} \text{ for some } x\}$.

Let $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$. The class \mathcal{L} can be iteratively learnt using 1-feedback. For the target language being L , a learner can determine $\min(L)$ (iteratively, in the limit) from a text for L . Furthermore, on an input $2x+b$, with $b \in \{0, 1\}$, the learner can use the feedback query $2x+1-b$. In case the learner ever gets the answer “yes”, it knows that the target language is from \mathcal{L}_2 and is $W_{\min(L)} \cup \{2x, 2x+1\}$. On the other hand, if the learner never gets the answer “yes” to a feedback query, then the target language is $W_{\min(L)}$.

Note that our iterative learner using 1-feedback, for \mathcal{L} , does not use negative counterexamples to conjectures. On the other hand, it can be shown that iterative learners, without feedback or memory, cannot learn the class \mathcal{L} even using negative counterexamples.

3 Some General Effects of Additional Information on NCIt-learning

In this section, we look at some known capabilities of **NCIt**-learners (established in [JK07]) and explore whether they hold when a learner has access to additional information.

It was shown in [JK07] that the capabilities of **NCIt**-learners exceed the capabilities of **Inflt**-learners. In this section, we show that if an **Inflt**-learner can store just one element seen so far, or can use just one feedback query, then it can sometimes learn more than any **NCEx**-learner (which can memorize the whole input seen so far!). However, total **Inflt**-learners having access to the maximal element seen so far still can be simulated by **NCIt**-learners having access to the maximal element seen so far.

An important result established in [JK07] is that **NCIt**-learners can infer any *indexed* class of recursive languages. However, it is also shown in [JK07] that, surprisingly, such **NCIt**-learners cannot learn indexed classes *class-preservingly* (cf.[LZZ08]), that is, using a numbering of languages containing exactly the target class (and no other languages). Still class-preserving learnability is important, as any natural hypotheses space for an indexed class is class-preserving. We show that **NCIt**-learners can learn indexed classes class-preservingly if they have access to the maximal element or the number of elements seen so far. However, adding the capability of using m feedback queries might not be enough to help an **NCIt**-learner to infer an indexed class class-preservingly.

3.1 Informants Versus Negative Counterexamples

As we already mentioned, **NCIt**-learners are more powerful than **Inflt**-learners ([JK07]). However, **Inflt**-learners using just one feedback can simulate any **NCIt**-learner, as the following theorem shows.

Theorem 9. *Any class $\mathcal{L} \in \mathbf{NCIt}$ can be **Inflt**-learnt using 1-feedback.*

Proof. Suppose M is an **NCIt**-learner for a class \mathcal{L} of languages. We define a learner M' , which **Inflt**-learns \mathcal{L} using 1-feedback, as follows.

Let $\sigma_0, \sigma_1, \dots$ denote an enumeration of all members of SEQ. Suppose L is the target language. We first give an informal description of how the learner M' works.

The learner M' searches for k, m_1, m_2 such that (i) for $j \leq m_1$, if $W_j \cap \bar{L} \neq \emptyset$, then $\min(W_{j, m_2} \cap \{x : x \leq m_2, x \notin L\}) = \min(W_j \cap \bar{L})$ (that is, least counterexamples to conjectures $j \leq m_1$, if any, can be determined based on simulation of W_j for m_2 steps and membership in L for $x \leq m_2$ (which can be determined using 1-feedback queries)), (ii) $\text{content}(\sigma_k) \subseteq \{x : x \leq m_2, x \in L\}$, (iii) $M(\sigma_k)$ halts within m_2 steps, and $M(\tau) \leq m_1$ for each $\tau \subseteq \sigma_k$, where the counterexamples given to M are the least ones, if any, and (iv) σ_k is a stabilizing sequence for M on the target language L , when the counterexamples given to M are the least counterexamples, if any.

Note that for such k, m_1, m_2 , the input language (if from the class \mathcal{L}) must be $W_{M(\sigma_k)}$, where the counterexamples to the conjectures of M are the least ones, if any. Furthermore, by a locking sequence argument, such k, m_1, m_2 exist as the learner M **NCIt**-learns L (here note that M learns the target language when it is given arbitrary (and, thus, the least) counterexamples to its conjectures).

Now, to search for σ_k, m_1, m_2 as above, the learner cycles through all triples $\langle k, m_1, m_2 \rangle$, discarding $\langle k, m_1, m_2 \rangle$ which are not suitable. To this end, the learner M' needs to determine membership for $x \leq m_2$ in the target language L (besides checking some other things). The conjectures of the learner would be of the form $R(\langle k, m_1, m_2 \rangle, S, r, b)$ and $P(i, \langle k, m_1, m_2 \rangle, S)$, where S is a subset of $\{x : x \leq m_2\}$, $r \leq m_2$, $b \in \{0, 1\}$, and R and P are 1–1 computable functions with disjoint range such that $W_{P(i, \langle k, m_1, m_2 \rangle, S)} = W_i$ (it doesn't matter what $W_{R(\langle k, m_1, m_2 \rangle, S, r, b)}$ is; these conjectures are used only to memorize the parameters). Note that R, P being 1–1 with disjoint range allows one to determine, from conjectures of M' , whether R or P was used to form the conjecture, as well as the corresponding parameters.

When the learner is determining whether the parameters k, m_1, m_2 satisfy (i)–(iv) above, it initially conjectures hypotheses of the form $R(\langle k, m_1, m_2 \rangle, S, r, b)$, which denotes that the learner has already determined membership for $x < r$ (given by S , which is equal to $L \cap \{x : x < r\}$), and is currently determining membership for r . Feedback queries of the form (r, b) , $b \in \{0, 1\}$ are used repeatedly to determine whether $r \in L$ or not. Once the learner has determined the membership for all $x \leq m_2$, it checks whether (ii) above holds, and then uses conjectures of the form $P(i, \langle k, m_1, m_2 \rangle, S)$ and the later inputs/feedback to check whether (i), (iii) and (iv) above hold.

We now proceed formally.

Description of M'

(a) $M'(\lambda) = R(0, \emptyset, 0, 0)$.

(b) On the previous conjecture being $R(\langle k, m_1, m_2 \rangle, S, r, b)$ and the input (y, a) , M' does the following:

(* Here the learner is trying to determine membership in the target language for $x \leq m_2$. Note that $(x, \chi_L(x))$, would eventually be answered yes in the feedback query. The exact value of the input (y, a) is not used by M' in this stage. *)

1. Query whether (r, b) has already appeared in the input.

2. If so, then

2.1. If $b = 1$, then let $S = S \cup \{r\}$.

2.2. If $r < m_2$, then conjecture $R(\langle k, m_1, m_2 \rangle, S, r + 1, 0)$.

(* That is, continue determining membership for $x \leq m_2$ *).

2.3. If $r = m_2$ and $\text{content}(\sigma_k) \subseteq S$, then

conjecture $P(0, \langle k, m_1, m_2 \rangle, S)$.

(* 0 above is just an arbitrary value; Step 4 below will determine the conjecture i of M on the input σ_k (assuming (i)–(iv) above hold). *)

Else, conjecture $R(\langle k, m_1, m_2 \rangle + 1, \emptyset, 0, 0)$ (that is, try the next possible triple $\langle k', m'_1, m'_2 \rangle = \langle k, m_1, m_2 \rangle + 1$).

Else, conjecture $R(\langle k, m_1, m_2 \rangle, S, r, 1 - b)$ (that is, test whether $(r, 1 - b)$ has appeared in the input).

(* Note that we repeatedly and alternately try to check if $(r, 0)$ or $(r, 1)$ has appeared in the input, until one of them appears. *)

(c) On the previous conjecture being $P(p, \langle k, m_1, m_2 \rangle, S)$ and the input (y, a) , M' does the following:

(* Note that a is ignored in this step. *)

3. For $j \leq m_1$, let $C_j = \min(W_{j, m_2} \cap \{x : x \leq m_2, x \notin S\})$, where if $W_{j, m_2} \cap \{x : x \leq m_2, x \notin S\} = \emptyset$, then we take $C_j = \#$.
4. Simulate M on the input σ_k , where counterexample to a conjecture j of M on a prefix τ of σ_k is given as C_j .

If the above simulation does not halt within m_2 steps or $M(\tau) > m_1$ for some $\tau \subseteq \sigma_k$, then conjecture $R(\langle k, m_1, m_2 \rangle + 1, \emptyset, 0, 0)$ (that is try the next possible triple $\langle k', m'_1, m'_2 \rangle$).

Else, let i be the conjecture of M on the input σ_k and proceed to step 5.

5. Suppose $y = \langle y', b', n' \rangle$.

Query whether (y', b') has appeared in the input before

If yes, then

- 5.1. If $b' = 0$, and for some $j \leq m_1$, $y' \in W_{j, n}$ and $y' < C_j$ (where we take $y' < \#$ for all y'), then conjecture $R(\langle k, m_1, m_2 \rangle + 1, \emptyset, 0, 0)$ (that is, try the next possible triple $\langle k', m'_1, m'_2 \rangle$).

Else, conjecture $P(i, \langle k, m_1, m_2 \rangle, S)$.

- 5.2. If $b' = 1$, and $M(\sigma_k \diamond y') \downarrow$ within n' steps and $M(\sigma_k \diamond y') \neq M(\sigma_k)$ (where the counterexample to a conjecture j is given as C_j), then conjecture $R(\langle k, m_1, m_2 \rangle + 1, \emptyset, 0, 0)$ (that is, try the next possible triple $\langle k', m'_1, m'_2 \rangle$).

Else, conjecture $P(i, \langle k, m_1, m_2 \rangle, S)$.

Else (i.e., (y', b') has not appeared in the input before), conjecture $P(i, \langle k, m_1, m_2 \rangle, S)$.

End

It is easy to verify that for any particular triple $\langle k, m_1, m_2 \rangle$, the learner initially uses conjectures of the form $R(\langle k, m_1, m_2 \rangle, \cdot, \dots, \cdot)$ to eventually correctly determine $S = L \cap \{x : x \leq m_2\}$ (here note that every pair $(x, \chi_L(x))$ eventually appears in the input, and thus eventually feed-back query for $(x, \chi_L(x))$ of step 1 is answered yes).

Now, for any particular triple $\langle k, m_1, m_2 \rangle$, once $S = L \cap \{x : x \leq m_2\}$ is determined by the learner M' , it computes $i = M(\sigma_k)$ in steps 3–4 (assuming (i), (iii) and (iv) hold; note that (ii) has been checked already in step 2.3). Violation of (i) is checked in step 5.1 (note that, for every $y' \in \mathbb{N}$, there are arbitrarily large n' such that $(\langle y', \chi_L(y'), n' \rangle, \chi_L(\langle y', \chi_L(y'), n' \rangle))$ appears in the input). Violation of (iv) is checked in step 5.2 (again note that, for every $y' \in \mathbb{N}$, there are arbitrarily large n' such that $(\langle y', \chi_L(y'), n' \rangle, \chi_L(\langle y', \chi_L(y'), n' \rangle))$ appears in the input). Violation of (iii) (assuming (i) holds) is detected in steps 3–4.

Thus, all triples which do not satisfy (i)–(iv) are eventually discarded by the learner. A correct triple which satisfies (i)–(iv) will never be discarded, and eventually the learner will output only $P(i, \langle k, m_1, m_2 \rangle, S)$, where $S = L \cap \{x : x \leq m_2\}$, and i is the conjecture of M on the input σ_k

when the negative counterexamples are the least, if they exist. It follows that M' **InfIt**-learns \mathcal{L} using 1-feedback. \square

Jain and Kinber [JK08] showed that **InfEx** – **NCEx** $\neq \emptyset$. We strengthen this result to show that there exists a class \mathcal{L} not in **NCEx** which can be identified by a **NCIt**-learner using 1-memory or 1-feedback.

A set X is *semi-recursive* (see [Joc68]) if there exists a recursive function f such that, (i) for all x, y , $f(x, y) \in \{x, y\}$ and (ii) if $x \in X$ or $y \in X$, then $f(x, y) \in X$. Equivalently, a set X is *semi-recursive* iff there exists a total computable ordering $<'$ on \mathbb{N} (which is very different from the standard ordering of natural numbers) such that $x <' y$ and $y \in X$ implies $x \in X$. It can be shown that there exist r.e., non-recursive but semi-recursive sets [Joc68]. Note that if a semi-recursive set X is not recursive, then, for each $y \notin X$, there exist infinitely many $z \notin X$ such that $z <' y$. To see this, suppose $y \notin X$. Then $y <' z$ implies $z \notin X$; thus, if there are only finitely many $z \notin X$ such that $z <' y$, then X is recursive.

Theorem 10. *There exists a class \mathcal{L} such that*

- (a) \mathcal{L} can be **InfIt**-identified by a 1-memory bounded (or 1-feedback) learner, and
- (b) $\mathcal{L} \notin \mathbf{NCEx}$.

Proof. Let A be a semi-recursive, nonrecursive r.e. set such that for every $x \in \mathbb{N}$, either both $2x$ and $2x + 1$ are in A or both $2x$ and $2x + 1$ are not in A . Such a set can be constructed by taking $A = \{2x + b : x \in B, b \in \{0, 1\}\}$, for any semi-recursive, r.e. but not recursive set B . Let $\mathcal{L} = \{A \cup \{y\} : y \in \mathbb{N}\}$.

We first show that $\mathcal{L} \notin \mathbf{NCEx}$. Suppose by way of contradiction that a learner M **NCEx**-identifies \mathcal{L} . Let σ be a locking sequence (for **NCEx**-learning) for M on A , where counterexamples are provided in some natural way. Then, for infinitely many $y \notin A$, on all initial segments $\tau \supseteq \sigma$ such that $\text{content}(\tau) \subseteq A \cup \{y\}$, $M(\tau) = M(\sigma)$. (This holds as, otherwise, \bar{A} would be r.e. (and thus A will be recursive) as, except for finitely many y , $y \in \bar{A}$ iff $M(\tau) \neq M(\sigma)$, for some $\tau \supseteq \sigma$ such that $\text{content}(\tau) \subseteq A \cup \{y\}$.) Let $y \notin A$ be such that (i) y has not been provided as counterexample to M on initial segments of σ , and (ii) $M(\tau) = M(\sigma)$ for all $\tau \supseteq \sigma$ such that $\text{content}(\tau) \subseteq A \cup \{y\}$. It follows that M does not **NCEx**-learn $A \cup \{y\}$, and, thus, does not **NCEx**-learn \mathcal{L} .

\mathcal{L} can be iteratively learnt from informant using 1-feedback as follows. Initially the learner conjectures A . On each input $(2x + a, b)$, with $a, b \in \{0, 1\}$, the learner queries if $(2x + 1 - a, 1 - b)$ has been seen in the input. If so, then the learner changes its mind to $A \cup \{2x + a'\}$, where $a' = a$, if $b = 1$ and $a' = 1 - a$, otherwise — and thereafter never changes its mind. It is easy to verify that the above learner will iteratively learn \mathcal{L} from informant using 1-feedback.

Let $<'$ be a computable total ordering over \mathbb{N} witnessing the semi-recursive nature of A . To iteratively learn from informant using 1-memory, the learner initially conjectures A . The learner memorizes the largest positive element (with respect to $<'$) seen in the input. That is, its memory consists of y such that $(y, 1)$ is in the input, and for all $(x, 1)$ in the input, $x <' y$. If and when the learner sees a negative datum x which is smaller (with respect to $<'$) than the memorized element (say y) at that time, it conjectures $A \cup \{y\}$, and then never changes its mind.

Now, if the input language is A , then the learner never sees a negative datum x such that $x <' y$, for some positive data y . Thus, the learner converges on the input language A to the conjecture A . On the other hand, if the input language is $A \cup \{y\}$, then y is the largest datum with respect to $<'$ that the learner sees. Thus, the memory of the learner is eventually y . As A is semi-recursive, but not recursive, there are infinitely many elements $x \notin A$ such that $x <' y$. Thus, eventually (after receiving y as positive datum), the learner sees an input $x <' y$ such that x is not in the input language. Then, the learner conjectures $A \cup \{y\}$ and never changes its mind thereafter. \square

Still, as the next theorem demonstrates, *total* **Inflt**-learners can be simulated by **NCIt**-learners if both have access to the maximal element seen so far. Here, total **Inflt**-learners are those learners which are defined on all, even, possibly, non-valid inputs — that is, even on data which does not represent a possible “(previous conjecture, a new input element, or the maximal element seen so far)”, in a valid learning process for a language in the class being learnt. For learning from informants, the maximal element seen so far in the input is taken as the maximal y such that $(y, 0)$ or $(y, 1)$ is present in the input given so far. It is open at present whether the following result holds if the totality assumption is removed.

Theorem 11. *Any class which is **Inflt** learnable using the maximal element seen so far by a total learner is also **NCIt**-learnable using the maximal element seen so far.*

Proof. Suppose an **Inflt** learner M , using the maximal element seen so far, for a class \mathcal{L} is given. Below $M(i, (w, b), x)$ denotes the output of M when the previous hypothesis is i , (w, b) is the current input, and x is the maximal element such that (x, b) , for some $b \in \{0, 1\}$, has been in the input provided to M so far.

On an input text T for a language $L \in \mathcal{L}$, the aim of the **NCIt**-learner M' (using maximal element seen so far) is to search for some initial segment σ of the canonical informant (for L) such that

- (A) For all $w \in L$, $w \geq |\sigma|$, $M(M(\sigma), (w, 1), w) = M(\sigma)$.
- (B) $\{w \geq |\sigma| : M(M(\sigma), (w, 0), w) \neq M(\sigma)\} \subseteq L$.

Note that any such σ would imply that $M(\sigma)$ is a grammar for the input language L . σ satisfying (A) and (B) above are called *good* (for L). Note that such good σ exist for all languages **Inflt** learnt by M (using maximal element seen so far as additional information).

The hypothesis of M' is of the form $P(\sigma)$ or $R(\sigma)$, or $Q(w, \sigma, S, m)$, where $w, m \in \mathbb{N}$, and S is a finite set. Here, $Q(w, \sigma, S, m)$ is a grammar for $\{w\}$, $P(\sigma)$ is a grammar for $W_{M(\sigma)}$, and $R(\sigma)$ is a grammar for $\{w \geq |\sigma| : M(M(\sigma), (w, 0), w) \neq M(\sigma)\}$. Furthermore, R, P, Q are 1–1 computable functions with disjoint range. In all the above conjectures, σ would be some appropriate initial segment of the canonical informant for the input language. $R(\sigma)$ is a conjecture of the learner when it is testing whether clause (B) above holds. $P(\sigma)$ is the hypothesis which the learner makes in a situation when it thinks that it has a potentially good σ for the input language (which satisfies (B) and satisfies (A) for the input seen so far). When the learner does not have a good σ , it needs to determine the membership of some w in the input language to extend σ — $Q(w, \sigma, S, m)$ is used for this purpose, and this phase determines the membership

for every x such that $|\sigma| \leq x \leq m$ (thus the lost data is recovered in this phase). During this membership determining phase, S would be a finite set of elements from the input, which the learner remembers separately, in order not to forget them due to memory limitation.

Initially, $M'(\lambda) = R(\lambda)$.

For other inputs M' behaves as follows.

Suppose the previous conjecture is $R(\sigma)$. If the new input is w , and the counterexample is not $\#$, then M' outputs $Q(|\sigma|, \sigma, \emptyset, \max(\{m, |\sigma| + 1\}))$, where m is the maximal element seen so far (note that in this case, (B) did not hold for the current σ). Similarly, if $w \neq \#$ and $M(M(\sigma), (w, 1), w) \neq M(\sigma)$, then M' outputs $Q(|\sigma|, \sigma, \emptyset, \max(\{m, |\sigma| + 1\}))$, where m is the maximal element seen so far (note that in this case (A) did not hold for the current σ). Otherwise, M' outputs $P(\sigma)$.

Suppose the previous conjecture is $P(\sigma)$. If the new input is w and the counterexample is not $\#$, then M' outputs $Q(|\sigma|, \sigma, \emptyset, \max(\{m, |\sigma| + 1\}))$, where m is the maximal element seen so far. Similarly, if $w \neq \#$, and $M(M(\sigma), (w, 1), w) \neq M(\sigma)$, then M' outputs $Q(|\sigma|, \sigma, \emptyset, \max(\{m, |\sigma| + 1\}))$, where m is the maximal element seen so far (note that in this case (A) does not hold for the current σ). Otherwise, M' continues with the hypothesis $P(\sigma)$.

Suppose the previous conjecture is $Q(x, \sigma, S, m)$. Let $b = 1$, if the counterexample is $\#$; $b = 0$ otherwise. (In this case x would be $|\sigma|$, and we will extend σ based on membership of x in L). If the new input is w , and $x < m$, then M' outputs $Q(x + 1, \sigma \diamond (x, b), S \cup \{w\} - \{\#\}, m)$. If $x = m$, then M' additionally checks if $M(\sigma \diamond (x, b)) \neq M(M(\sigma \diamond (x, b)), (y, 1), y)$ for some $y \in S \cup \{w\} - \{\#\}$, $y > |\sigma|$; if so then M' outputs $Q(x + 1, \sigma \diamond (x, b), \emptyset, \max(S \cup \{w\} - \{\#\}) + m + 1)$. (The previous step checked if σ satisfies (A) for all the data stored in S). Otherwise, M' outputs $R(\sigma \diamond (x, b))$.

Note that all σ used in the hypotheses of the form $R(\sigma)$, $Q(w, \sigma, S, m)$, and $P(\sigma)$ are initial segments of the canonical informant for the input language. Also note that either the hypotheses of the learner M' converge, or σ (as in the hypotheses) is unbounded in length.

Also note that there exists a good τ which is an initial segment of the canonical informant, as M identifies the input language on the canonical informant. Fix one such good segment τ for the input language. Furthermore, all initial segments of the canonical informant, which extend τ , are also good. It is then easy to verify that once the σ , as in the hypotheses of M' , extends τ , M' will eventually output $P(\sigma')$ for some extension σ' of σ such that σ' is an initial segment of the canonical informant of the input language (for this, note that if the previous hypothesis of M' was $R(\sigma)$ or $P(\sigma)$, then it will next output $P(\sigma)$ and will not change its mind thereafter; if its previous hypothesis is of the form $Q(i, \sigma, S, m)$, then once value of i reaches m , the learner will output $R(\sigma')$ for some extension σ' of σ , where σ' is an initial segment of the canonical informant for the input language). It follows that eventually the learner M' will output $P(\sigma)$ where σ is good, and M' then never changes its mind. It thus follows that M' identifies L . \square

Jain and Kinber [JK07] had shown that $\mathbf{NCIt} - \mathbf{InfIt} \neq \emptyset$. Below, we generalize this to show that some \mathbf{NCIt} -learnable class cannot be \mathbf{InfIt} -learnt even if the \mathbf{InfIt} -learner uses n -memory. Intuitively, the reason for this is that if the target language misses out finitely many elements from a set (such as \mathbb{N} used in the proof below), then these finitely many elements can be determined by using negative counterexamples, even if one has forgotten certain earlier

data. However, an **Inflt**-learner (even with n -memory) cannot determine such elements, if it has forgotten some earlier data.

Theorem 12. *There exists a class \mathcal{L} such that*

- (a) $\mathcal{L} \in \mathbf{NCIt}$, and
- (b) *for every n , \mathcal{L} cannot be learnt by an n -memory bounded **Inflt**-learner.*

Proof. Let $E = \{2x : x \in \mathbb{N}\}$. Let $\mathcal{L} = \{L : L \subseteq E \text{ and } W_{\min(L)/2} = L\} \cup \{L : \mathbb{N} - L \text{ is finite}\}$.

To show that \mathcal{L} can be **NCIt**-learnt, consider the following learner. Until the learner sees an element in the input which is not in E , it conjectures $i/2$ for i being the minimal element seen so far. If and when the learner sees an element outside E in the input, it starts by outputting a conjecture for \mathbb{N} , and then continues by outputting conjectures for $\mathbb{N} - S$, where S is the set of negative counterexamples it sees. It is easy to verify that the above learner would **NCIt**-learn the class \mathcal{L} .

Suppose by way of contradiction that \mathcal{L} can be **Inflt**-learnt using n -memory by a learner M . Then, by implicit use of Kleene's recursion theorem [Rog67], there exists an e such that W_e can be defined as follows. Note that M is defined on all inputs, as it identifies all cofinite sets. Below by an "initial information segment" we mean an initial segment of the canonical informant of some language.

Initially, enumerate $2e$ into W_e and let σ_0 be an initial information segment such that $\text{content}(\sigma_0) = \{(x, 0) : x < 2e\} \cup \{(2e, 1)\}$. Go to stage 0.

Stage s

1. Search for an initial information segment σ_{s+1} which extends σ_s such that $\{(x, 1) : x \in \text{content}(\sigma_{s+1})\} \subseteq E$ and the conjectures of M on σ_s and σ_{s+1} are different.
2. If and when such a σ_{s+1} is found, enumerate $\{x : (x, 1) \in \text{content}(\sigma_{s+1})\}$ into W_e and go to stage $s + 1$.

End Stage s

Clearly, if there are infinitely many stages then $I = \bigcup_s \sigma_s$ is an informant for W_e on which M diverges. On the other hand, if stage s starts but never ends, then one can find two initial information segments τ and τ' extending σ_s such that both τ and τ' are of same length but different in content and M has the same conjecture and memory after seeing τ or τ' . (There are 2^m different possible τ which extend σ_s and satisfy $|\tau| = |\sigma_s| + 2m$ and $\{x : (x, 1) \in \text{content}(\tau)\} \subseteq E$. However, the memory on such τ has at most $(2|\sigma_s| + 4m + 1)^n$ many possibilities. Thus, for a large enough m , one can find τ and τ' as required.) It follows that if one considers the cofinite languages $\mathbb{N} - \{x : (x, 0) \in \text{content}(\tau)\}$ and $\mathbb{N} - \{x : (x, 0) \in \text{content}(\tau')\}$, then both are in \mathcal{L} . However, M would converge to the same conjecture (if any) on both τI and $\tau' I$, where $\text{content}(I) = \{(x, 1) : x \geq |\tau|\}$. It follows that M fails to identify at least one of $\mathbb{N} - \{x : (x, 0) \in \text{content}(\tau)\}$ and $\mathbb{N} - \{x : (x, 0) \in \text{content}(\tau')\}$. \square

3.2 Indexed Classes

Unlike the case of **NCIt**-learnability (without access to additional information), class-preserving learnability of indexed classes can be achieved if an **NCIt**-learner has access to the maximal element or the number of elements seen so far. Here a learner is class-preserving (for learning a class \mathcal{L}) if it only conjectures decision procedures for languages in the class \mathcal{L} .

Theorem 13. (a) *Every indexed class can be **NCIt**-identified by a class-preserving learner when it is given the maximal element seen so far as additional information.*

(b) *Every indexed class can be **NCIt**-identified by a class-preserving learner when it is given the number of elements seen so far as additional information.*

Proof. (a) Suppose \mathcal{L} is an indexed class, and L_0, L_1, \dots is its listing where $x \in L_i$ can be effectively determined in x and i . Let $L_i[m]$ denote $\{x \in L_i : x \leq m\}$. The conjectures of the learner would be of the form: $p(j, S, X)$, where $p(j, S, X)$ is a decision procedure for L_j , and S, X are finite sets with some properties.

Suppose T is an input text for a language L , where $T(n) = x_n$. Inductively, if $p(j_n, S_n, X_n)$ is the output after $T[n]$ has been seen, then the following invariants will hold.

- (A) For each $j \in S_n$, $L_j \subseteq L$, and $X_n \subseteq L$.
- (B) $\text{content}(T[n]) \subseteq X_n \cup \bigcup_{j \in S_n} L_j$,
- (C) For all $j < j_n$, $L_j \neq L$.
- (D) If $j_n \notin S_n$, then either $n = 0$ or $j_n = j_{n-1} + 1$.
- (E) $X_n \subseteq X_{n+1}$, $S_n \subseteq S_{n+1}$, $j_n \leq j_{n+1}$.

Initially, $M(\lambda) = (0, \emptyset, \emptyset)$. The learner on the input $p(j_n, S_n, X_n)$ and the new element x_n , the counterexample y_n , and the maximal element m seen so far, does the following:

- (i) If $y_n = \#$, then $S_{n+1} = S_n \cup \{j_n\}$; otherwise $S_{n+1} = S_n$.
- (ii) If $(X_n \cup \{x_n\} \cup \bigcup_{j \in S_n} L_j[m]) - \{\#\} \subseteq L_{j_n}$, and $y_n = \#$, then $j_{n+1} = j_n$, $X_{n+1} = X_n$.

Otherwise, $j_{n+1} = j_n + 1$ and $X_{n+1} = X_n \cup \{x_n\} - \{\#\}$.

It is easy to verify that the invariants are satisfied. Furthermore, j_n never goes beyond the minimal index for L (see invariant (C)). Thus, the sequence of j_n converges, as well as S_n and X_n converge (as $X_{n+1} \neq X_n$ implies $j_{n+1} \neq j_n$, and $S_n \subseteq \{j : j \leq j_n\}$, and using invariants (D) and (E)). Moreover, the last conjecture is correct by (A) and (B), and using $(X_n \cup \{x_n\} \cup \bigcup_{j \in S_n} L_j[m]) - \{\#\} \subseteq L_{j_n}$ from the clause (ii) (as there is no further mind change).

(b) The only change is in (ii) above, which is replaced by: (m below denotes the number of elements seen so far by the learner)

- (ii) If the first m elements in $(X_n \cup \{x_n\} \cup \bigcup_{j \in S_n} L_j) - \{\#\}$ are included in L_{j_n} , and $y = \#$, then $j_{n+1} = j_n$, $X_{n+1} = X_n$. Otherwise, $j_{n+1} = j_n + 1$ and $X_{n+1} = X_n \cup \{x_n\} - \{\#\}$.

The rest of the proof is similar to the part (a), and we omit the details. \square

Still, any n feedback queries might not help to achieve class-preserving learnability of indexed classes by **NCIt**-learners.

The following result can be proved by using the technique of Theorem 32 in [JK07], where it is shown that some indexed class cannot be **NCIt**-learnt with respect to any class preserving

hypothesis space. The necessary modification of the proof in [JK07] makes sure that the construction preserves the answers for the feedback queries in building the diagonalizing language. We omit the details.

Theorem 14. *There exists an indexed class which cannot be learnt by a class-preserving **NCIt**-learner using n -feedback.*

4 Hierarchy of n -Feedback and n -Memory Learners

In this section, we show that, in the context of **NCIt**-learnability, $n + 1$ stored input elements seen and $n + 1$ feedback queries provide more capability than n stored input elements seen and, respectively, n feedback queries. Note that, on the negative sides of both results, neither **NCIt**-learners storing just up to n elements seen, nor **NCIt**-learner using just up to n feedback queries can be helped even if they have access to the maximal element and the number of elements seen so far. On the other hand, learners witnessing the positive sides of both results do not need access to negative counterexamples (refuting conjectures containing data in excess of the target language).

Intuitively, the following technique is used to prove Theorems 15 and 16 below. Consider the classes

$$\mathcal{L}_1 = \{W_e : W_e \subseteq \{\langle e, x \rangle : x \in \mathbb{N}\}\}, \text{ and}$$

$$\mathcal{L}_2 = \{W_e \cup \{\langle e, x \rangle\} : W_e \in \mathcal{L}_1, W_e \text{ is finite and } x > \max(\{x' : \langle e, x' \rangle \in W_e\})\}.$$

Then, $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$ cannot be **InfEx**-learnt, though each of \mathcal{L}_1 and \mathcal{L}_2 can be **TextEx**-learnt. This result can be proven in a way similar to the non-union theorem [BB75]. We then modify the above class as follows:

$$\mathcal{L}'_1 = \{W_e : W_e \subseteq \{\langle e, j, x \rangle : x \in \mathbb{N}\} \text{ and for each } x, \text{ there exists at most one } j \geq 1 \text{ such that } \langle e, j, x \rangle \in W_e\}, \text{ and}$$

$$\mathcal{L}'_2 = \{W_e \cup \{\langle e, j_1, x \rangle, \langle e, j_2, x \rangle\} : W_e \in \mathcal{L}'_1, W_e \text{ is finite, } x > \max(\{x' : (\exists j)[\langle e, j, x' \rangle \in W_e]\}), \text{ and } 1 \leq j_1 < j_2 \leq n + 2\}.$$

Then, $\mathcal{L}'_1 \cup \mathcal{L}'_2$ becomes learnable as long as one can “detect” whether there are j_1, j_2 such that $1 \leq j_1 < j_2 \leq n + 1$, and $\langle e, j_1, x \rangle$ and $\langle e, j_2, x \rangle$ are in the target language. This can be done if one is allowed $n + 1$ feedback queries (by checking on the input $\langle e, j, x \rangle$, $1 \leq j \leq n + 2$, whether any of $\langle e, j', x \rangle$ has been seen in the input, for $1 \leq j' \leq n + 2$, $j' \neq j$). However, such a check cannot be done by an n -feedback learner. Negative counterexamples, the maximal element seen so far, and the number of elements seen so far, cause some minor complications. In the construction below elements of the form $\langle e, 0, x \rangle$ are used mainly to handle the additional information “maximal element seen so far”.

The proof for the memory-bounded hierarchy (Theorem 16) uses a similar idea, but instead of using one versus two elements of the form $\langle e, j, x \rangle$ in \mathcal{L}'_1 and \mathcal{L}'_2 , respectively (for a particular x), one uses “ $\leq n + 1$ elements or $n + 2$ elements satisfying some property” versus “ $n + 2$ elements not satisfying the property” (the particular property chosen is that the sum of such j ’s is prime; however various other similar properties could be used). The complication regarding the special property being satisfied for $n + 2$ elements (rather than just using “ $\leq n + 1$ elements” versus “ $n + 2$ ”

elements”) was needed to be able to diagonalize against the additional information “number of elements seen so far”.

Note that our classes used for demonstrating the advantages of extra feedback/memory are somewhat more transparent than the ones used in [CJLZ99] for the feedback and memory hierarchy; same applies to the corresponding proofs — at least partially, perhaps, due to the fact that corresponding diagonalizations in [CJLZ99] must work against learners vacillating in the limit between a finite number of correct conjectures, whereas it is not necessary in our setting.

We now proceed formally.

Theorem 15. *Fix $n \in \mathbb{N}$. There exists a class \mathcal{L} such that*

(a) *\mathcal{L} can be iteratively learnt by an $(n + 1)$ -feedback learner.*

(b) *\mathcal{L} cannot be **NCIt**-learnt using n -feedback queries even if the maximal element and the number of elements in the input seen so far is given to the learner as additional information.*

Proof. Let

$$\begin{aligned} \mathcal{L}_1 = \{L : (\exists e)[\ \emptyset \subset L \subseteq \{\langle e, j, x \rangle : j, x \in \mathbb{N}\}, \\ W_e = L \text{ and} \\ \text{for all } x, [\text{card}(\{j : \langle e, j, x \rangle \in L, j \geq 1\}) \leq 1] \\]]\} \end{aligned}$$

$$\begin{aligned} \mathcal{L}_2 = \{L : (\exists e, x, j, j' : 1 \leq j < j' \leq n + 2)[\ W_e \in \mathcal{L}_1, \\ x > \max(\{x' : (\exists j' \geq 1)[\langle e, j', x' \rangle \in W_e]\}) \text{ and} \\ L = W_e \cup \{\langle e, j, x \rangle, \langle e, j', x \rangle\} \\]]\}. \end{aligned}$$

Let $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$.

It is easy to verify that \mathcal{L} can be learnt using $n + 1$ feedback queries. The learner can easily determine e such that the input language is a subset of $\{\langle e, j, x \rangle : j, x \in \mathbb{N}\}$. Initially, on the first non $\#$ input, the learner outputs e (padded so that the learner can recognize that it is in this phase). For every further input of the form $\langle e, j, x \rangle$ such that $1 \leq j \leq n + 2$, the learner queries if the earlier data contains any of the elements in $\{\langle e, j', x \rangle : 1 \leq j' \leq n + 2, j \neq j'\}$. If so, then the learner outputs a grammar for $W_e \cup \{\langle e, j, x \rangle, \langle e, j', x \rangle\}$, for the $j' \neq j$ such that $\langle e, j', x \rangle$ belongs to the input seen so far, and does not change its mind any further.

Now suppose, by way of contradiction, that a learner M **NCIt**-identifies \mathcal{L} using n feedback queries (along with the maximal element as well as the number of elements seen so far). Then, by implicit use of Kleene’s recursion theorem, there exists an e such that W_e may be defined as follows.

Initially, W_e contains $\langle e, 0, 0 \rangle$. Let W_e^s denote W_e defined by the beginning of stage s . Let σ_0 be a sequence with the content $\{\langle e, 0, 0 \rangle\}$. Let σ_s denote the initial segment constructed before stage s (it will be the case that $W_e^s = \text{content}(\sigma_s)$). Furthermore, $f_s(i)$ will be a function denoting

counterexamples given to the learner M on its conjecture i (in the simulation at stage s). It will be the case that the range of f_s (except for $\#$) is a subset of E_s — which we will bar from belonging to W_e to maintain the validity of any negative counterexamples given. Initially, $f_0(i) = \#$, for all i , and $E_0 = \emptyset$. Let x_s denote the least number such that $W_e^s \cup E_s \subseteq \{\langle e, j, x \rangle : x < x_s\}$. Go to stage 0.

Stage s

1. Let $m > x_s$ be a large enough number such that $\langle e, 0, m \rangle > \max(\text{content}(\sigma_s) \cup \{\langle e, j, x \rangle : x_s \leq x \leq x_s + 1 \text{ and } 1 \leq j \leq n + 2\})$. Enumerate $\langle e, 0, m \rangle$ in W_e , and let $\tau = \sigma_s \diamond \langle e, 0, m \rangle$.
2. Simulate M by giving counterexamples according to f_s . Dovetail steps 3 and 4 until one of them succeeds. If step 3 succeeds before step 4, if ever, then go to step 5. If step 4 succeeds before step 3, if ever, then go to step 6. Here we assume that step 3 has some priority in the sense that if it can succeed for $t \leq s$, then it succeeds first, with σ being the shortest one for which such $t \leq s$ exists.
3. (* This step checks if any of the counterexamples provided, as given by f_s , is potentially not a least counterexample. *).

Search for a $\sigma \subseteq \tau$ and a t such that $W_{M(\sigma),t} - \text{content}(\tau) \neq \emptyset$ and $\min(W_{M(\sigma),t} - \text{content}(\tau)) \neq f_s(M(\sigma))$.

4. (* This step tries to force a mind change by M . *)

Search for a j, j', x such that $1 \leq j, j' \leq n + 2$, $j \neq j'$, $x_s \leq x \leq x_s + 1$ and

(a) $M(\tau \diamond \langle e, j, x \rangle) \downarrow \neq M(\tau) \downarrow$ or

(b) $M(\tau \diamond \langle e, j, x \rangle \langle e, j', x \rangle) \downarrow \neq M(\tau) \downarrow$, where M on previous conjecture $M(\tau \diamond \langle e, j, x \rangle)$ and new data $\langle e, j', x \rangle$ did not query $\langle e, j, x \rangle$.

5. Let

$$\sigma_{s+1} = \tau,$$

$$f_{s+1}(M(\sigma)) = \min(W_{M(\sigma),t} - \text{content}(\tau)),$$

$$f_{s+1}(i) = f_s(i), \text{ for } i \neq M(\sigma),$$

$$W_e^{s+1} = W_e \text{ enumerated until now.}$$

$$E_{s+1} = E_s \cup \{f_{s+1}(M(\sigma))\}, \text{ and}$$

$$x_{s+1} = \text{the least number such that } W_e^{s+1} \cup E_{s+1} \subseteq \{\langle e, j, x \rangle : x < x_{s+1}\}.$$

Go to stage $s + 1$.

6. In case (a) let $\sigma_{s+1} = \tau \diamond \langle e, j, x \rangle$.

In case (b) let $\sigma_{s+1} = \tau \diamond \langle e, j'', x' \rangle \langle e, j', x \rangle$, where j'', x' is such that $1 \leq j'' \leq n + 2$ and $x \neq x'$, $x_s \leq x' \leq x_s + 1$ and $\langle e, j'', x' \rangle$ is not queried by $M(\tau \diamond \langle e, j, x \rangle \langle e, j', x \rangle)$.

Let $W_e^{s+1} = \text{content}(\sigma_{s+1})$ and update $E_{s+1} = E_s$, $f_{s+1} = f_s$ and $x_{s+1} = \text{the least number such that } W_e^{s+1} \cup E_{s+1} \subseteq \{\langle e, j, x \rangle : x < x_{s+1}\}$.

Go to stage $s + 1$

End stage s

Now, if there are infinitely many stages, then $W_e \in \mathcal{L}_1$, and $T = \bigcup_s \sigma_s$ is a text for W_e . As M identifies W_e , $M(T)$ converges. Thus, for a large enough stage s , step 3 would not succeed

anymore (as the least counterexamples would have been found by then). Thus, step 4 succeeds infinitely often, and M does not converge on T , a contradiction to the assumption that $M(T)$ converges.

Thus, there are only finitely many stages. Suppose stage s starts but does not end. Hence the counterexamples as given by f_s on initial segments of τ (as in stage s) are correct (and least ones, whenever they exist). Let $\langle e, j', x_s \rangle, \langle e, j, x_s \rangle$, $j \neq j'$ with $1 \leq j, j' \leq n+2$, be such that M on the previous conjecture $M(\tau \diamond \langle e, j, x_s \rangle)$ and new input $\langle e, j', x_s \rangle$, does not query $\langle e, j, x_s \rangle$. Note that $M(\tau) \downarrow = M(\tau \diamond \langle e, j, x_s \rangle) \downarrow = M(\tau \diamond \langle e, j, x_s \rangle \langle e, j', x_s \rangle) \downarrow$, and $W_{M(\tau)}$ either does not enumerate any element outside $\text{content}(\tau)$, or the least such element is $f_s(M(\tau))$ (by non-success of steps 3 and 4), where $f_s(M(\tau))$ is different from $\langle e, j, x_s \rangle, \langle e, j', x_s \rangle$ (by definition of x_s). Thus, M does not **NCIt** learn with n feedback queries the language $W_e \cup \{\langle e, j, x_s \rangle, \langle e, j', x_s \rangle\}$, which belongs to \mathcal{L}_2 . \square

Theorem 16. *Let $n \in \mathbb{N}$. There exists a class \mathcal{L} such that*

(a) \mathcal{L} can be iteratively learnt using $(n+1)$ -memory.

(b) \mathcal{L} cannot be learnt by an **NCIt**-learner using n -memory, even if the learner is given the number of elements and the maximal element seen so far as additional information.

Proof. Let

$$\begin{aligned} \mathcal{L}_1 = \{L : (\exists e)[\ & \emptyset \subset L \subseteq \{\langle e, j, x \rangle : j, x \in \mathbb{N}\}, \\ & W_e = L \text{ and} \\ & \text{for all } x[\text{card}(\{j : \langle e, j, x \rangle \in L, j \geq 1\}) \leq n+1 \text{ or} \\ & \quad [\text{card}(\{j : \langle e, j, x \rangle \in L, j \geq 1\}) = n+2 \text{ and} \\ & \quad \sum_{\langle e, j, x \rangle \in L} j \text{ is a prime number}] \\ & \quad]]\} \end{aligned}$$

$$\begin{aligned} \mathcal{L}_2 = \{L : (\exists e, x)[\ & W_e \in \mathcal{L}_1, \\ & x > \max(\{x' : \langle e, j, x' \rangle \in W_e, j' \geq 1\}), \\ & L = W_e \cup \{\langle e, j, x \rangle : j \geq 1, \langle e, j, x \rangle \in L\} \text{ and} \\ & [\text{card}(\{j : \langle e, j, x \rangle \in L, j \geq 1\}) = n+2 \text{ and} \\ & \quad \sum_{\langle e, j, x \rangle \in L} j \text{ is not a prime number}] \\ & \quad]]\} \end{aligned}$$

Let $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$.

It is easy to verify that \mathcal{L} can be learnt using $n+1$ memory. The learner can easily determine e such that the input language is a subset of $\{\langle e, j, x \rangle : j, x \in \mathbb{N}\}$. Initially, on the first non

input, the learner outputs e (padded so that it can recognize that it is in this phase). The memory of the learner always consists of all elements of the form $\langle e, j, x \rangle$ seen so far (unless it exceeds $n + 1$), for the maximal x such that, for some j' , $\langle e, j', x \rangle$ is seen in the input so far. If and when a learner sees an input $\langle e, j, x \rangle$, $j \geq 1$, such that it has memorized $n + 1$ distinct elements from the set $\{\langle e, j', x \rangle : j' \geq 1, j' \neq j\}$, and $j + \sum_{\langle e, j', x \rangle \in \text{memory}} j'$ is not a prime number, then it outputs $W_e \cup \{\langle e, j, x \rangle\} \cup \{\langle e, j', x \rangle : \langle e, j', x \rangle \text{ in memory}\}$, and never changes its mind thereafter.

Now suppose, by way of contradiction, that a learner M **NCIt**-identifies \mathcal{L} using n -memory (along with the maximal element as well as the number of elements seen so far). Then, by implicit use of Kleene's recursion theorem, there exists an e such that W_e may be defined as follows.

Initially, W_e contains $\langle e, 0, 0 \rangle$. Let W_e^s denote W_e defined by the beginning of stage s . Let σ_0 be a sequence with content $\{\langle e, 0, 0 \rangle\}$. Let σ_s denote the initial segment constructed before stage s (it will be the case that $W_e^s = \text{content}(\sigma_s)$). Furthermore, $f_s(i)$ will be the function denoting counterexamples given to the learner M on its conjecture i (in the simulation at stage s). It will be the case that the range of f_s (except for #) is a subset of E_s — which we will bar from belonging to W_e to maintain the validity of any negative counterexample given. Initially, $f_0(i) = \#$, for all i , and $E_0 = \emptyset$. Let x_s denote the least number such that $W_e^s \cup E_s \subseteq \{\langle e, j, x \rangle : x < x_s\}$. Go to stage 0.

Stage s

1. Let $w > n + 1$ be a large enough number such that $(w + \text{card}(\text{content}(\sigma_s)) + 2)^n < \binom{w}{n+1}$. Let w' be such that for all distinct $c, c' \leq (n + 1) * 2^w$, there exists a p with $2^w < p \leq w'$ such that $c + p$ is a prime, but $c' + p$ is not a prime. Let $m > x_s$ be such that $\langle e, 0, m \rangle > \max(\text{content}(\sigma_s) \cup \{\langle e, j, x_s \rangle : 1 \leq j \leq w'\})$. Enumerate $\langle e, 0, m \rangle$ in W_e , and let $\tau = \sigma_s \diamond \langle e, 0, m \rangle$.
2. Simulate M by giving counterexamples according to f_s . Dovetail steps 3 and 4 until one of them succeeds. If step 3 succeeds before step 4, if ever, then go to step 5. If step 4 succeeds before step 3, if ever, then go to step 6. Here we assume that step 3 has some priority in the sense that if it can succeed for $t \leq s$, then it succeeds first, with σ being the shortest for which such $t \leq s$ exists.
3. Search for a $\sigma \subseteq \tau$ and a t such that $W_{M(\sigma), t} - \text{content}(\tau) \neq \emptyset$ and $\min(W_{M(\sigma), t} - \text{content}(\tau)) \neq f_s(M(\sigma))$.
4. Search for a τ' such that either
 - (a) $\text{content}(\tau') - \text{content}(\tau) \subseteq \{\langle e, j, x_s \rangle : 1 \leq j \leq w'\}$, and $\text{card}(\text{content}(\tau') - \text{content}(\tau)) \leq n + 1$, and $M(\tau') \neq M(\tau)$ or
 - (b) $\text{content}(\tau') - \text{content}(\tau) \subseteq \{\langle e, j, x_s \rangle : 1 \leq j \leq w'\}$, and $\text{card}(\text{content}(\tau') - \text{content}(\tau)) = n + 2$, and $\sum_{\langle e, j, x \rangle \in \text{content}(\tau') - \text{content}(\tau)} j$ is a prime number, and $M(\tau') \neq M(\tau)$.
5. Let
$$\begin{aligned} \sigma_{s+1} &= \tau, \\ f_{s+1}(M(\sigma)) &= \min(W_{M(\sigma), t} - \text{content}(\tau)), \\ f_{s+1}(i) &= f_s(i), \text{ for } i \neq M(\sigma), \end{aligned}$$

$W_e^{s+1} = W_e$ enumerated until now.
 $E_{s+1} = E_s \cup \{f_{s+1}(M(\sigma))\}$, and
 x_{s+1} = the least number such that $W_e^{s+1} \cup E_{s+1} \subseteq \{\langle e, j, x \rangle : x < x_{s+1}\}$.
 Go to stage $s + 1$.
 6. Let $\sigma_{s+1} = \tau'$,
 Let $W_e^{s+1} = \text{content}(\sigma_{s+1})$ and update $E_{s+1} = E_s$, $f_{s+1} = f_s$ and x_{s+1} = the least number
 such that $W_e^{s+1} \cup E_{s+1} \subseteq \{\langle e, j, x \rangle : x < x_{s+1}\}$.
 Go to stage $s + 1$
 End stage s

Now, if there are infinitely many stages, then $W_e \in \mathcal{L}_1$, and $T = \bigcup_s \sigma_s$ is a text for W_e . As M identifies W_e , $M(T)$ converges. Thus for large enough stage s , step 3 would not succeed anymore (as the least counterexamples would have been found by then). Thus, step 4 succeeds infinitely often, and M does not converge on T , a contradiction to the assumption that $M(T)$ converges.

Thus, there are only finitely many stages. Suppose stage s starts but does not end. Hence the counterexamples as in f_s on initial segments of τ (as in stage s) are correct. Let the parameters below be as in stage s . For each set S of $n + 1$ elements in $\{2^j : 1 \leq j \leq w\}$, let τ_S be such that $\tau \subseteq \tau_S$ and $\text{content}(\tau_S) - \text{content}(\tau) = S$. Let $q_S = \sum_{\langle e, j, x \rangle \in S} j$. Let S, S' be distinct sets of $n + 1$ elements in $\{2^j : 1 \leq j \leq w\}$ such that memory of $M(\tau_S)$ and memory of $M(\tau_{S'})$ are same. Note that there exist such distinct S, S' by the hypothesis on w . Furthermore, $q_S \neq q_{S'}$. Now, let p be such that $2^w < p \leq w'$, and $q_S + p$ is a prime number, but $q_{S'} + p$ is not a prime number. Then, as $M(\tau) = M(\tau_{S'}) = M(\tau_S) = M(\tau_S \diamond \langle e, p, x_s \rangle^\infty)$, we also have $M(\tau) = M(\tau_{S'} \diamond \langle e, p, x_s \rangle^\infty)$. Furthermore, $W_{M(\tau)}$ either does not enumerate any element outside $\text{content}(\tau)$, or the least such element is $f_s(M(\tau))$ (by non-success of step 3), which is different from $\langle e, p, x_s \rangle$ (by the definition of x_s). Thus, $M(\tau_{S'} \diamond \langle e, p, x_s \rangle^\infty)$ does not converge to a grammar for $\text{content}(\tau_{S'}) \cup \{\langle e, p, x_s \rangle\}$, which is a member of \mathcal{L}_2 . \square

5 Advantages of Different Types of Additional Information Over Other Types

In this section we study tradeoffs between different types of additional information in the context of **NCIt**-learnability.

5.1 Comparison of Feedback and Memory Bounded Learning

Results of this subsection significantly strengthen corresponding results given in [CJLZ99]. Namely, they demonstrate that, in the context of **NCIt**-learnability, just one stored input element can provide more than any n feedback queries (even if, in addition, the learner has access to the maximal element and the number of elements seen so far), and, conversely, one feedback query can do more than any n stored input elements seen so far (and, additionally, the maximal element and the number of elements seen so far). Moreover, the iterative learners witnessing

the positive sides of these results do not use negative counterexamples to conjectures containing extra elements.

Theorem 17. *There exists a class \mathcal{L} such that*

- (a) \mathcal{L} can be **TxtIt**-identified by a 1-memory bounded learner, but
- (b) \mathcal{L} cannot be **NCIt**-learnt using n -feedback (even if the learner is given the maximal element and the number of elements seen so far as additional information).

Proof. For each s , let \mathcal{L}_s denote the class \mathcal{L} used in the proof of Theorem 15. Let $\mathcal{S} = \bigcup_s \mathcal{L}_s$. Then, by Theorem 15, \mathcal{S} cannot be **NCIt**-learnt using n -feedback. However, \mathcal{S} can be **TxtIt**-identified using 1-memory: Initially the learner outputs e (padded), so that $\langle e, \cdot, \cdot \rangle$ is the first element in the input. The learner always memorizes $\langle e, j', x \rangle$ for the largest x such that $\langle e, j', x \rangle$ belongs to the input for some $j' \geq 1$ (as long as there is only one such corresponding j'). Now if some $\langle e, j, x \rangle$ appears in the input for some $j \geq 1$, with $\langle e, j', x \rangle$ in the memory, where $j' \neq j$, then, the learner outputs the grammar for $W_e \cup \{\langle e, j, x \rangle, \langle e, j', x \rangle\}$, and never changes its mind thereafter. \square

Theorem 18. *There exists a class \mathcal{L} such that*

- (a) \mathcal{L} can be **TxtIt**-identified by a 1-feedback learner, but
- (b) \mathcal{L} cannot be **NCIt**-learnt by an n -memory bounded learner (even if the learner is given the maximal element and the number of elements in the input so far as additional information).

Proof. Let

$$\begin{aligned} \mathcal{L}_1 = \{L : (\exists e)[\ & \emptyset \subset L \subseteq \{\langle e, k, j, x \rangle : k, j, x \in \mathbb{N}\}, \\ & W_e = L, \\ & \text{for all } k > 0, j > 0, x, [\langle e, 1, j, x \rangle \in L \text{ iff } \langle e, k, j, x \rangle \in L] \text{ and} \\ & \text{for all } x, [\text{card}(\{j : j \geq 1, \langle e, 1, j, x \rangle \in L\}) = 0 \text{ or} \\ & \quad [\langle e, 1, j, x \rangle \in L \text{ for a unique } j \geq 1, \text{ and } \pi_1^2(j) \notin L]] \\ &]]\} \end{aligned}$$

$$\begin{aligned} \mathcal{L}_2 = \{L : (\exists e, x)[\ & W_e \in \mathcal{L}_1, \\ & x > \max(\{x' : \langle e, k, j, x' \rangle \in W_e, k \in \mathbb{N}, j \geq 1\}) \text{ and} \\ & [\langle e, 1, j, x \rangle \in L \text{ for a unique } j \geq 1, \\ & \quad \pi_1^2(j) \in L, \\ & \quad D_{\pi_2^2(j)} \subseteq \{\langle e, k, 0, x \rangle : k, x \in \mathbb{N}\} \text{ and} \\ & \quad L = W_e \cup D_{\pi_2^2(j)} \cup \{\langle e, k, j, x \rangle : k \geq 1\}] \\ &]]\}. \end{aligned}$$

Let $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$.

Using a proof similar to the proof of Theorem 16, it can be shown that \mathcal{L} cannot be **NCIt**-learnt using n -memory (even if the learner is given the maximal element and the number of elements seen so far as additional information). However, the above \mathcal{L} is learnable using 1-feedback query, as initially the learner can output conjecture e (actually, a padded version — to recognize that it is in this phase), and whenever it receives input of the form $\langle e, k, j, x \rangle$, for $j \geq 1$, $k \geq 1$, it can query whether $\pi_1^2(j)$ is in the text seen so far. If so, then it can change its mind to output grammar for $W_e \cup D_{\pi_2^2(j)} \cup \{\langle e, k', j, x \rangle : k' \geq 1\}$, and never change its mind thereafter. \square

Another class (which is simpler to state), which can be used for a proof of Theorem 18 would be:

$$\begin{aligned}\mathcal{L}_1 &= \{L : (\exists e)[W_e = L \subseteq \{\langle e, x \rangle : x \in \mathbb{N}\}] \text{ and for all } x, \text{card}(L \cap \{\langle e, 2x \rangle, \langle e, 2x + 1 \rangle\}) \leq 1\}. \\ \mathcal{L}_2 &= \{L : (\exists e, x)[W_e \in \mathcal{L}_1, L = W_e \cup \{\langle e, 2x \rangle, \langle e, 2x + 1 \rangle\}]\}. \\ \mathcal{L} &= \mathcal{L}_1 \cup \mathcal{L}_2.\end{aligned}$$

The diagonalization proof, though similar to the proof of Theorem 16, needs a bit more modification compared to what we needed for the class used in the current proof of Theorem 18.

5.2 Advantages of Using Maximal Element/Number of Elements

Results of this subsection demonstrate various advantages that **NCIt**-learners can get while using the maximal element or/and the number of elements seen so far as additional information.

We begin with two simple useful propositions. The following proposition works if memory, instead of being a set, is allowed to be a multiset (when updating the memory, if a new input element is greater than the current maximal one, the learner must replace the old maximal by the new one, however, the learner may also decide to store a separate copy of the new element — for reasons different from it being maximal, so that it would not be sacrificed when a new greater element appeared). It is open at present whether this proposition holds if memory is just a set, as in the current paper.

Proposition 19. *Any n -memory bounded learner with the maximal element seen so far as additional information can be simulated by an $(n + 1)$ -memory bounded learner by using the extra memory for the maximal element seen so far, as long as the memory of the learner is considered as a multi-set, rather than just a set.*

Proposition 20. *An **NCIt**-learner can learn finite sets when given the number of elements or the maximal element seen so far, and only negative counterexamples (no positive data is needed). Thus, the learner can, for example, do this even when it forgets some of the elements it has received due to some earlier phase.*

Proof. For the following without loss of generality, assume that the number of elements and the maximal element seen so far is as in the limit for the input text.

For the case of the number of elements seen so far, the learner tries, one by one, each possible finite set of cardinality m (where m is the number of elements seen so far), until it gets no counterexample.

For the case of the maximal element seen so far, the learner could just check for each x smaller than the maximal element whether x belongs to the input language (this can be done by using the conjecture $\{x\}$ — x is in the input language iff the conjecture for $\{x\}$ does not get a counterexample). That will determine the input language. \square

Our next result shows that adding access to the maximal element seen so far increases the learning capability of **NCIt**-learners storing up to n input elements seen so far. Moreover, a learner witnessing the positive side of the result does not need access to negative counterexamples refuting conjectures containing data in excess of the language to be learned.

The proof of the following theorem uses a modification of the diagonalizing class used in the proof of Theorem 16, though some of the components of that class are not needed. The class \mathcal{L}_1 contains languages which have at most n elements of the form $\langle e, 1, j, x \rangle$, for each x , and the class \mathcal{L}_2 contains languages which have $n + 1$ elements of the form $\langle e, 1, j, x \rangle$, for maximal x such that $\langle e, 1, j', x \rangle$ is present in the input for some j' — in the former case, W_e is the grammar for the language, and in the latter case the language is finite. This allows for easy learnability using n -memory and the maximal element seen so far — using n -memory one can check which of the above cases applies, and using the maximal element seen so far one can learn finite sets, even if one has forgotten some data. However, n -memory bounded learner cannot learn the class as it cannot recall some forgotten elements, unless it memorizes some large elements — which in turn hurts its learning of the class \mathcal{L}_1 .

Theorem 21. *There exists a class \mathcal{L} such that*

- (a) \mathcal{L} can be **TextIt**-identified by an n -memory bounded learner with the maximal element seen so far as additional input, but
- (b) \mathcal{L} cannot be **NCIt**-learnt by an n -memory bounded learner.

Proof. Let

$$\begin{aligned} \mathcal{L}_1 = \{L : (\exists e)[\, & W_e = L, \\ & \emptyset \subset L \subseteq \{\langle e, d, r, z \rangle : r, z \in \mathbb{N}, d > 0\} \text{ and} \\ & \text{for all } x, [\text{card}(\{j : \langle e, 1, j, x \rangle \in L\}) \leq n] \\ &]\}. \end{aligned}$$

$$\begin{aligned} \mathcal{L}_2 = \{L : (\exists e, x', y)[\, & L \subseteq \{\langle e, d, r, z \rangle : d, r, z \in \mathbb{N}\}, \\ & L \text{ is finite and} \\ & [\text{for all } x < x'[\text{card}(\{j : \langle e, 1, j, x \rangle \in L\}) \leq n, \\ & \quad \max(\{x : \langle e, 1, j, x \rangle \in L\}) = x' \text{ and} \\ & \quad \text{card}(\{j : \langle e, 1, j, x' \rangle \in L\}) = n + 1]] \\ &]\}. \end{aligned}$$

Let $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$.

It is easy to verify that \mathcal{L} can be learnt using n -memory plus the maximal element seen so far as additional information. The learner first finds an e such that $\langle e, d, y, z \rangle$ is in the input language. Then the learner remembers the $\langle e, 1, j, x \rangle$ in its memory, where x is maximized. The learner continues to output e , until it discovers that the input language contains $n + 1$ elements of the form $\langle e, 1, j', x \rangle$, for the maximal x such that some element of the form $\langle e, 1, j, x \rangle$ is in the input (for this, the learner needs to memorize only first n such elements — when $n + 1$ -th such element arrives in the input, it would then already know that the target language contains $n + 1$ such elements). It then switches to learning finite sets using the maximal element (see Proposition 20).

Now suppose, by way of contradiction, that a learner M **NCIt**-identifies \mathcal{L} using at most n -memory. Note that, for all e , M is defined on all inputs σ such that $\text{content}(\sigma) \subseteq \{\langle e, d, r, z \rangle : d, r, z \in \mathbb{N}\}$, and for all z , $\text{card}(\{j : \langle e, 1, j, z \rangle \in \text{content}(\sigma)\}) \leq n$ (as any finite such set can be extended to be a member of \mathcal{L}_2 by adding, for a large enough z' , $\langle e, 1, j, z' \rangle$, for $n + 1$ different j). Now, by implicit use of Kleene's recursion theorem, there exists an e such that W_e may be defined as follows.

Initially, W_e contains $\langle e, 2, 0, 0 \rangle$. Let W_e^s denote W_e defined by the beginning of stage s . Let σ_0 be a sequence with content $\{\langle e, 2, 0, 0 \rangle\}$. Let σ_s denote the initial segment constructed before stage s (it will be the case that $W_e^s = \text{content}(\sigma_s)$). Furthermore, $f_s(i)$ will be the function denoting counterexamples given to the learner M on its conjecture i (in the simulation at stage s). It will be the case that the range of f_s (except for $\#$) is a subset of E_s — which we will bar from belonging to W_e to maintain the validity of any negative counterexample given.

Initially, $f_0(i) = \#$, for all i , and $E_0 = \emptyset$. Let x_s denote the least number such that $W_e^s \cup E_s \subseteq \{\langle e, d, j, x \rangle : d, j \in \mathbb{N}, x < x_s\}$. Go to stage 0.

Stage s

1. Simulate M by giving counterexamples according to f_s . Dovetail steps 2 and 3 until one of them succeeds. If step 2 succeeds before step 3, if ever, then go to step 4. If step 3 succeeds before step 2, if ever, then go to step 5. Here we assume that step 2 has some priority in the sense that if it can succeed for $t \leq s$, then it succeeds first, with σ being the shortest for which such $t \leq s$ exists.
2. Search for a $\sigma \subseteq \sigma_s$ and a t such that $W_{M(\sigma),t} - \text{content}(\sigma_s) \neq \emptyset$ and $\min(W_{M(\sigma),t} - \text{content}(\sigma_s)) \neq f_s(M(\sigma))$.
3. Search for a $\tau \supseteq \sigma_s$ such that (a) $\text{content}(\tau) - \text{content}(\sigma_s) \subseteq \{\langle e, d, y, z \rangle : y, z \in \mathbb{N}, d > 0\}$ and for all x , $[\text{card}(\{j : \langle e, 1, j, x \rangle \in \text{content}(\tau)\}) \leq n]$, and (b) $M(\sigma_s) \neq M(\tau)$.
4. Let

$$\begin{aligned}
\sigma_{s+1} &= \sigma_s, \\
f_{s+1}(M(\sigma)) &= \min(W_{M(\sigma),t} - \text{content}(\tau)), \\
f_{s+1}(i) &= f_s(i), \text{ for } i \neq M(\sigma), \\
W_e^{s+1} &= W_e \text{ enumerated until now.} \\
E_{s+1} &= E_s \cup \{f_{s+1}(M(\sigma))\}, \text{ and} \\
x_{s+1} &= \text{the least number such that } W_e^{s+1} \cup E_{s+1} \subseteq \{\langle e, d, j, x \rangle : d, j \in \mathbb{N}, x < x_{s+1}\}.
\end{aligned}$$

Go to stage $s + 1$.

5. Let $\sigma_{s+1} = \tau$,
 Let $W_e^{s+1} = \text{content}(\sigma_{s+1})$, $E_{s+1} = E_s$, $f_{s+1} = f_s$, $x_{s+1} =$ the least number such that
 $W_e^{s+1} \cup E_{s+1} \subseteq \{\langle e, d, j, x \rangle : d, j \in \mathbb{N}, x < x_{s+1}\}$.
 Go to stage $s + 1$
 End stage s

Now, if there are infinitely many stages, then $W_e \in \mathcal{L}_1$, and $T = \bigcup_s \sigma_s$ is a text for W_e . As M identifies W_e , $M(T)$ converges. Thus for a large enough stage s , step 2 would not succeed anymore (as the least counterexamples would have been found by then). Thus, step 3 succeeds infinitely often, and M does not converge on T , a contradiction to the assumption that $M(T)$ converges.

Thus, there are only finitely many stages. Suppose stage s starts but does not end. Hence the counterexamples as in f_s on initial segments of τ (as in stage s) are correct. Let the parameters below be as in stage s . For each set S of n elements, let τ_S be such that $\sigma_s \subseteq \tau_S$ and $\text{content}(\tau_S) - \text{content}(\sigma_s) = \{\langle e, 1, j, x_s \rangle : j \in S\}$.

Now, suppose there exists an S (of size n) such that for infinitely many y , $M(\tau_S \diamond \langle e, 2, 0, y \rangle)$ has the same memory as $M(\tau_S)$. Let j' be such that $j' \notin S$. Then, consider M 's behaviour on $\tau_S \diamond \langle e, 1, j', x_s \rangle^\infty$. If it does not converge, then it does not learn $\text{content}(\tau_S \diamond \langle e, 1, j', x_s \rangle)$ from the text $\tau_S \diamond \langle e, 1, j', x_s \rangle^\infty$ (where the counterexamples provided are the least ones). Otherwise, let X be the set of counterexamples provided to M on $\tau_S \diamond \langle e, 1, j', x_s \rangle^\infty$, and let y be such that $\langle e, 2, 0, y \rangle$ is not in the set of counterexamples provided nor $\langle e, 2, 0, y \rangle \in E_s$, and $M(\tau_S \diamond \langle e, 2, 0, y \rangle)$ has same memory as $M(\tau_S)$. Then, M fails to identify at least one of $\tau_S \diamond \langle e, 1, j', x_s \rangle^\infty$ and $\tau_S \diamond \langle e, 2, 0, y \rangle \langle e, 1, j', x_s \rangle^\infty$, as it converges to the same conjecture on both these texts.

Otherwise, for all S (of size n), for all but finitely many y , $M(\tau_S \diamond \langle e, 2, 0, y \rangle)$ has different memory than $M(\tau_S)$.

Thus, by taking such S as a subset of size n of $\{1, 2, \dots, w\}$, for a large enough w , we will have that, for two such S, S' , for all but finitely many y , memory is the same after seeing $\tau_S \diamond \langle e, 2, 0, y \rangle$ or after seeing $\tau_{S'} \diamond \langle e, 2, 0, y \rangle$ (as the memory could either be remembering $\langle e, 2, 0, y \rangle$ or not, and some set of the size at most $n-1$, due to change in memory). Let $j \in S - S'$ and y be large enough (satisfying above) such that $\langle e, 2, 0, y \rangle \notin E_s$. Then, $M(\tau_{S'} \diamond \langle e, 2, 0, y \rangle \langle e, 1, j, x_s \rangle^\infty) = M(\tau_S \diamond \langle e, 2, 0, y \rangle \langle e, 1, j, x_s \rangle^\infty) = M(\sigma_s)$, and $W_{M(\sigma_s)}$ either enumerates an element in E_s , or does not enumerate any element outside $\text{content}(\sigma_s)$. Thus, it fails to identify $\tau_{S'} \diamond \langle e, 2, 0, y \rangle \langle e, 1, j, x_s \rangle^\infty$, which is a member of \mathcal{L}_2 . \square

Our next result demonstrates the advantages of **NCIt**-learner getting the maximal element compared to it getting the number of elements seen so far. However, we were not able to achieve a result of similar strength while faring the *number of elements* seen so far against the maximal element seen so far as additional information. Whether it is possible, remains open. A partial solution to this problem (for iterative learners — not using negative counterexamples to conjectures) is given in Theorem 27.

Theorem 22. *There exists a class \mathcal{L} such that*

- (a) \mathcal{L} can be **TextIt**-identified when the learner is provided the maximal element seen so far as additional information, but

(b) \mathcal{L} cannot be **NCIt**-identified when the learner is given the number of elements in the input as additional information.

Proof. Let

$$\begin{aligned} \mathcal{L}_1 = \{L : 0 \notin L \text{ and} \\ & [[L \text{ is finite and } W_{\pi_1^3(\max(L))} = L] \text{ or} \\ & [L \text{ is infinite and for some } e \in \mathbb{N}, \\ & \text{for all but finitely many } x \in L, [\pi_1^3(x) = e \text{ and } W_e = L]] \\ &]\}. \end{aligned}$$

$$\begin{aligned} \mathcal{L}_2 = \{L : 0 \in L \text{ and} \\ & [[L \text{ is finite and } W_{\pi_2^3(\max(L))} = L] \text{ or} \\ & [L \text{ is infinite and for some } e \in \mathbb{N}, \\ & \text{for all but finitely many } x \in L [\pi_2^3(x) = e, \text{ and } W_e = L]]]\}. \end{aligned}$$

Let $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$.

It is easy to verify that a learner can iteratively learn \mathcal{L} using the maximal element seen so far as additional information.

Now suppose by way of contradiction that M **NCIt**-identifies \mathcal{L} using the number of elements seen so far. Note that M must converge on all inputs, as every finite data has an extension in \mathcal{L}_2 . The aim of the construction below is to first try to force infinitely many mind changes for M on some language W_e from \mathcal{L}_1 . If this is not possible, then W_e would be finite, and one chooses a stabilizing sequence σ_s for M on $\{\langle e, x, y \rangle : x, y \in \mathbb{N}\} - E_s$, such that $\text{content}(\sigma_s) = W_e$ (E_s is a set of elements which we exclude from the target language due to negative counterexamples provided to M in the construction). Then, the construction tries to see if it can force infinitely many mind changes by M for some $W_{e'} \supseteq W_e$ such that $W_{e'} \in \mathcal{L}_2$: this search is done carefully so that if infinitely many mind changes cannot be forced, then one obtains an appropriate finite set extending $W_{e'}$ (the set $W_{e'} \cup \{d\}$, with d being of the form $\langle e, \cdot, \cdot \rangle$, in the construction below) which is not identified by M . We now proceed formally.

By implicit use of Kleene's recursion theorem, there exists an $e > 0$ such that W_e may be described as follows. Note that $\langle e, x, y \rangle > 0$, for all x, y by our assumption on pairing functions. Initially, let W_e contain $\langle e, 0, 0 \rangle$. Let W_e^s denote W_e defined by the beginning of stage s . Let σ_0 be a sequence with content $\{\langle e, 0, 0 \rangle\}$. Go to stage 0. Let σ_s denote the initial segment constructed before stage s (it will be the case that $W_e^s = \text{content}(\sigma_s)$). Furthermore, $f_s(i)$ will be the function denoting counterexamples given to the learner M on its conjecture i (in the simulation at stage s). It will be the case that the range of f_s (except for $\#$) is a subset of E_s — which we will bar from belonging to W_e to maintain the validity of any negative counterexample given. Initially, $f_0(i) = \#$, for all i , and $E_0 = \emptyset$. Go to stage 0.

Stage s

1. Simulate M by giving counterexamples according to f_s . Dovetail steps 2 and 3 until one of them succeeds. If step 2 succeeds before step 3, if ever, then go to step 4. If step 3 succeeds before step 2, if ever, then go to step 5. Here we assume that step 2 has some priority in the sense that if it can succeed for $t \leq s$, then it succeeds first, with σ being the shortest one for which such $t \leq s$ exists.
2. Search for a $\sigma \subseteq \sigma_s$ and a $t \in \mathbb{N}$ such that $W_{M(\sigma),t} - \text{content}(\sigma_s) \neq \emptyset$ and $\min(W_{M(\sigma),t} - \text{content}(\sigma_s)) \neq f_s(M(\sigma))$.
3. Search for a $\tau \supseteq \sigma_s$ such that $\text{content}(\tau) \subseteq \{\langle e, x, y \rangle : x, y \in \mathbb{N}\} - E_s$ and $M(\tau) \neq M(\sigma_s)$.
4. Let

$$\begin{aligned}\sigma_{s+1} &= \sigma_s \\ f_{s+1}(M(\sigma)) &= \min(W_{M(\sigma),t} - \text{content}(\sigma_s)), \\ f_{s+1}(i) &= f_s(i), \text{ for } i \neq M(\sigma), \\ E_{s+1} &= E_s \cup \{f_{s+1}(M(\sigma))\}, \\ W_e^{s+1} &\text{ be } W_e \text{ enumerated upto now,}\end{aligned}$$

Go to stage $s + 1$.

5. Enumerate $\text{content}(\tau)$ into W_e .
Let W_e^{s+1} be W_e enumerated upto now.
Let σ_{s+1} be an extension of τ such that $\text{content}(\sigma_{s+1}) = W_e^{s+1}$.
Let $E_{s+1} = E_s$, $f_{s+1} = f_s$.
Go to stage $s + 1$

End stage s

Now, if there are infinitely many stages, then $W_e \in \mathcal{L}_1$, and $T = \bigcup_s \sigma_s$ is a text for W_e . As M identifies W_e , $M(T)$ converges. Thus, for a large enough stage s , step 2 would not succeed anymore (as the least counterexamples would have been found by then). Thus, step 3 succeeds in all but finitely many stages, and M does not converge on T , a contradiction to the assumption that $M(T)$ converges. Hence, $M(T)$ diverges and thus M does not identify W_e .

Now consider the case that there are only finitely many stages. Suppose stage s starts but does not end. Hence the counterexamples as in f_s on initial segments of τ (as in stage s) are correct. Moreover, $W_{M(\sigma_s)} \subseteq \text{content}(\sigma_s)$ or $W_{M(\sigma_s)}$ contains an element in E_s . Furthermore, M does not change its mind on any extension τ of σ_s such that $\text{content}(\tau) \subseteq \{\langle e, x, y \rangle : x, y \in \mathbb{N}\} - E_s$.

Now, by implicit use of Kleene's recursion theorem, there exists an $e' > \max(\text{content}(\sigma_s) \cup E_s)$ such that $W_{e'}$ may be described as follows. In the construction below, we will start from stage $s + 1$, to maintain variable name consistency with the stages above.

$W_{e'}^t$ denotes $W_{e'}$ constructed before stage t . Initially, let $W_{e'}^{s+1} = W_e^s \cup \{\langle e, e', 0 \rangle, 0\}$ and let $\sigma_{s+1} = \sigma_s \diamond \langle e, e', 0 \rangle \diamond 0$. We let $\tau_{s+1} = \sigma_s \diamond d \diamond 0$, where we leave the element d unspecified for now (it will be the case that $\pi_1^3(d) = e$). We will assume that d is different from any other element encountered in the construction below. (This is for the ease of presentation; the analysis later will ensure that any such properties used would hold). Let $f_{s+1} = f_s$ and $E_{s+1} = E_s$.

It will be the case that, for $t \geq s + 1$, $W_{e'}^t = \text{content}(\sigma_t)$. However, we will not necessarily have that $\sigma_t \subseteq \sigma_{t+1}$. This is because in some cases, we rearrange some elements of σ_t to form

σ_{t+1} (similar rearrangement will apply to τ_{t+1} too). It will, however, be the case that, in the case of infinitely many stages s , for each n $\lim_{t \rightarrow \infty} \sigma_t(n)$ would converge. Furthermore, we will have that $\tau_{s+1} \subseteq \tau_t$ and $\sigma_{s+1} \subseteq \sigma_t$, for all $t \geq s + 1$.

σ_t, τ_t would satisfy the following invariants (for $t \geq s + 1$).

(A) $|\sigma_t| = |\tau_t|$, and $\text{card}(\text{content}(\sigma_t)) = \text{card}(\text{content}(\tau_t))$.

(B) For all $m \leq |\sigma_t|$, $M(\sigma_t[m]) = M(\tau_t[m])$, where the counterexamples given to M are based on f_t .

(C) For all m with $|\tau_{s+1}| \leq m \leq |\tau_t|$, $\text{content}(\tau_t[m]) - \text{content}(\sigma_t[m]) = \{d\}$.

(D) For all m with $|\sigma_{s+1}| \leq m < |\sigma_t|$, either (D1) $[\sigma_t(m) = \tau_t(m) \text{ and } \text{card}(\text{content}(\sigma_t[m+1]) - \text{content}(\tau_t[m+1])) = 1]$, or (D2) $[\tau_t(m) = \#, \sigma_t(m) \notin (\text{content}(\sigma_t[m]) \cup \text{content}(\tau_t[m]))]$, $M(\sigma_t[m]) = M(\tau_t[m]) = M(\sigma_t[m+1]) = M(\tau_t[m+1])$ (where the counterexamples are given based on f_t), $\tau_t(m+1) = \sigma_t(m+1)$, $\sigma_t(m+1) \in \text{content}(\sigma_t[m]) - \text{content}(\tau_t[m])$, and $\text{content}(\sigma_t[m+2]) - \text{content}(\tau_t[m+2]) = \{\sigma_t(m)\}$ (note that in this case we will have that $m+1 < |\sigma_t|$).

Thus, at all initial segments of σ_t and τ_t of the length $m \geq |\sigma_{s+1}|$, we maintain that $\sigma_t[m]$ has exactly one extra element compared to $\tau_t[m]$, except that (as in D2) we may temporarily add a new element to σ_t (as $\sigma_t(m)$), and then add back the earlier difference between $\sigma_t[m]$ and $\tau_t[m]$ to both σ_t and τ_t (as $\sigma_t(m+1)$ and $\tau_t(m+1)$). This is done when the learner M does not seem to make a mind change from input being $\sigma_t[m]$ (or $\tau_t[m]$) to input being either $\sigma_t[m] \diamond \#$ or $\sigma_t[m] \diamond \sigma_t(m)$.

Note that the invariants are clearly satisfied before stage $s + 1$.

Go to stage $s + 1$.

Stage t

1. Simulate M by giving counterexamples according to f_t . Dovetail steps 2 and 3 until one of them succeeds. If step 2 succeeds before step 3, if ever, then go to step 4. If step 3 succeeds before step 2, if ever, then go to step 5. Here we assume that step 2 has some priority in the sense that if it can succeed for $k \leq t$, then it succeeds first, with σ being the shortest one for which such $k \leq t$ exists.

2. Search for a $\sigma \subseteq \sigma_t$ and a $k \in \mathbb{N}$ such that $W_{M(\sigma),k} - \text{content}(\sigma_t) \neq \emptyset$ and $\min(W_{M(\sigma),k} - \text{content}(\sigma_t)) \neq f_t(M(\sigma))$.

Note that we take the shortest σ which satisfies the search condition (for the k found).

3. Let n be large enough such that $\langle e, e', n \rangle > \max(\text{content}(\sigma_t) \cup E_t)$. Let z be such that $z \in \text{content}(\sigma_t) - \text{content}(\tau_t)$.

Search for a $\gamma \in \{\#, \langle e, e', n \rangle, \langle e, e', n \rangle \diamond z, \langle e, e', n \rangle \diamond z \diamond \#\}$ such that $M(\sigma_t \gamma) \neq M(\sigma_t)$.

Here we take the shortest γ among the above possibilities.

4. Note that we take the shortest σ which satisfies the search condition of step 2 (for the k found). Thus, by using invariant (D) we have that $\text{card}(\text{content}(\sigma) - \text{content}(\tau_t[\lceil \sigma \rceil])) = 1$.

Let

$$f_{t+1}(M(\sigma)) = \min(W_{M(\sigma),t} - \text{content}(\sigma_t)),$$

$$f_{t+1}(i) = f_t(i), \text{ for } i \neq M(\sigma),$$

$$E_{t+1} = E_t \cup \{f_{t+1}(M(\sigma))\},$$

$$W_{e'}^{t+1} \text{ be } W_{e'} \text{ enumerated upto now.}$$

Let α be such that $\text{content}(\alpha) = W_{e'}^{t+1} - \text{content}(\sigma)$

Let $\sigma_{t+1} = \sigma \diamond \alpha$.

Let $\tau_{t+1} = \tau_t[\lceil \sigma \rceil] \diamond \alpha$.

(* Note that we have rearranged σ_t to form σ_{t+1} to ensure that the new counterexample given to $M(\sigma)$, does not violate invariant (D). *)

Go to stage $t + 1$.

5. Enumerate $\text{content}(\tau)$ into $W_{e'}$.

Let $W_{e'}^{t+1}$ be $W_{e'}$ enumerated upto now.

Let $\sigma_{t+1} = \sigma_t \diamond \gamma$.

Let $\tau_{t+1} = \tau_t \diamond \gamma'$, where $\gamma' = \gamma$, if $\gamma \in \{\#, \langle e, e', n \rangle\}$; otherwise, γ' is obtained by replacing $\langle e, e', n \rangle$ in γ by $\#$.

Let $E_{t+1} = E_t$, $f_{t+1} = f_t$.

Go to stage $t + 1$.

End stage t

Invariants can be shown by induction. It is easy to see that the invariants (B) and (C) are maintained. We now argue for invariants (A) and (D) for σ_{t+1}/τ_{t+1} based on which step succeeds in stage t .

If step 3 succeeds in stage t , then clearly, (A) holds for σ_{t+1}/τ_{t+1} , as either $\text{content}(\sigma_{t+1}) - \text{content}(\tau_{t+1}) = \text{content}(\sigma_t) - \text{content}(\tau_t) = \{z\}$ or $\text{content}(\sigma_{t+1}) - \text{content}(\tau_{t+1}) = \{\langle e, e', n \rangle\}$, based on any choice of γ . Furthermore, (D1) clearly holds if $\gamma = \#$ or $\gamma = \langle e, e', n \rangle$. If $\gamma = \langle e, e', n \rangle z$ (similar argument holds for $\gamma = \langle e, e', n \rangle z \#$) then (D1) holds for $m = |\sigma_t|$, and $m = |\sigma_t| + 2$ and (D2) holds for $m = |\sigma_t| + 1$, as $M(\sigma_t) = M(\tau_t)$, $M(\sigma_t \#) = M(\tau_t \#)$, $M(\sigma_t \diamond \langle e, e', n \rangle) = M(\tau_t \diamond \langle e, e', n \rangle)$, $M(\sigma_t \diamond \langle e, e', n \rangle z) = M(\tau_t \diamond \# z)$ (by M being based only on the number of elements and the last element seen) and $M(\sigma_t) = M(\sigma_t \#) = M(\sigma_t \diamond \langle e, e', n \rangle)$, by choice of γ not being $\#$ or $\langle e, e', n \rangle$.

If step 2 succeeds in stage t , then note that σ chosen must be such that for $m = |\sigma| - 1$, (D1) was satisfied for σ_t/τ_t , as otherwise by (D2) we would have that $M(\sigma[\lceil \sigma \rceil - 1]) = M(\sigma)$ (for counterexamples given according to f_t), and we would have that $\sigma[\lceil \sigma \rceil - 1]$ would have been chosen at step 2. Thus, (A) holds for $\sigma_{t+1} = \sigma \alpha$ and $\tau_{t+1} = \tau_t[\lceil \sigma \rceil] \alpha$, as α only contained elements not in σ . Similarly, (D1) holds for each $m \geq |\sigma|$, as $\text{content}(\sigma) - \text{content}(\tau_t[\lceil \sigma \rceil])$ is not a member of α . Also, clearly for $m < |\sigma| - 1$, (D) holds for σ_{t+1}/τ_{t+1} as it held for σ_t/τ_t .

Thus, all the invariants are maintained.

Now suppose there are infinitely many stages. Then it is easy to argue by induction on m that, $\lim_{t \rightarrow \infty} \sigma_t(m)$ and $\lim_{t \rightarrow \infty} \tau_t(m)$ converge (since the counterexample provided eventually reaches the least possible value). Let the texts formed by taking these converged values be respectively T_σ and T_τ . We first argue that M does not converge on T_σ . If not, then clearly beyond some stage, only step 3 can succeed, as the counterexamples to conjectures of M on T_σ would have converged beyond some point. But then, by construction, $M(T_\sigma)$ would diverge.

Now suppose stage t starts but does not end. But then, $M(\sigma_t) = M(\sigma_t \diamond \langle e, e', n \rangle) = M(\sigma_t \diamond \#) = M(\sigma_t \diamond \langle e, e', n \rangle z) = M(\sigma_t \diamond \langle e, e', n \rangle z \#)$, by non-success of step 3. As, $M(\sigma_t) = M(\tau_t)$, $M(\sigma_t \#) = M(\tau_t \#)$, $M(\sigma_t \diamond \langle e, e', n \rangle z) = M(\tau_t \diamond \# z)$, and $M(\sigma_t \diamond \langle e, e', n \rangle z \#) = M(\tau_t \diamond \# z \#)$

(which all hold as M depends only on the number of elements and the last element of the input), we have that $M(\tau_t) = M(\tau_t \diamond \#) = M(\tau_t \diamond \#z) = M(\tau_t \diamond \#z\#)$. Furthermore, $W_{M(\tau_t)} \subseteq \text{content}(\sigma_t)$, or the counterexample as given by $f_t(M(\tau_t))$ does not belong to $\text{content}(\sigma_t)$. It follows that M does not identify any language $\text{content}(\sigma_t) \cup \{d\}$, as long as $d \in \{\langle e, x, y \rangle : x, y \in \mathbb{N}\} - (\text{content}(\sigma_t) \cup E_t)$. This holds as $M(\tau_t \#z\#\infty)$ converges to $M(\tau_t \#z) = M(\sigma_t)$, and $W_{M(\sigma_t)} \subseteq \text{content}(\sigma_t)$. Now, by implicit use of Kleene's recursion theorem, there exists an e'' such that, for $d = \langle e, e'', n \rangle$, $d > \max(\text{content}(\sigma_t) \cup E_t)$ and $W_{e''} = \text{content}(\sigma_t) \cup \{d\} \in \mathcal{L}_2$. Theorem follows. \square

Note that \mathcal{L}_1 and \mathcal{L}_2 used in the proof above are each in **NCIt**. To see this for \mathcal{L}_1 , note that a learner can store in its padding all e such that $e = \pi_1^3(x)$, for some x in the input language $L \in \mathcal{L}_1$, as there is only a finite number of such values e . Now, the learner can use unions of the languages W_e for such values e as its conjectures, excluding from such a union any W_e that contains an element in excess of the target language L (provided to the learner as a counterexample to its conjecture). Obviously, the final union of such languages W_e will be the target language L .

Our next result shows the advantages of having the maximal element or the number of elements seen so far against having n -feedback. It is open at present if one can strengthen the result to show that there exists a class which can be **NCIt**-learnt using the maximal element (respectively, the number of elements) seen so far but cannot be learnt using n -feedback and the number of elements (respectively maximal element and/or m -memory) seen so far. The main idea used in the proof of the following result is that, using the maximal element or the number of elements seen so far, one can learn finite sets using only negative counterexamples, without using any positive data (thus even when one has forgotten some past data), see Proposition 20. However, n -feedback cannot be used to recover past data once forgotten.

Theorem 23. *There exists a class \mathcal{L} such that*

- (a) \mathcal{L} can be **NCIt**-identified using the maximal element or the number of elements seen so far as additional information, but,
- (b) for all n , \mathcal{L} cannot be **NCIt**-learnt using n -feedback.

Proof. Let $\mathcal{L}_1 = \{\{\langle e+1, x \rangle : x \in W_e\} : W_e \neq \emptyset\}$,
 $\mathcal{L}_2 = \{D : D \text{ is finite and } (\exists y)[\langle 0, y \rangle \in D]\}$, and
 $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$.

It is easy to verify that \mathcal{L} can be iteratively learnt using the maximal element or the number of elements seen so far as additional information. If and when the learner sees $\langle 0, y \rangle$ in the input, for some y , the learner would switch to learning finite sets (see Proposition 20). Otherwise, if the learner sees some input (and only inputs) of the form $\langle e+1, x \rangle$, the learner outputs a grammar for $\{\langle e+1, x \rangle : x \in W_e\}$.

To see that the above class \mathcal{L} cannot be learnt using n -feedback by **NCIt**-learner, suppose, by way of contradiction, that M does so. Then, by implicit use of Kleene's recursion theorem [Rog67], there exists an e such that W_e may be described as follows.

Initially, W_e contains 0. Let W_e^s denote W_e defined by the beginning of stage s . Let σ_0 be a sequence with content $\{\langle e+1, 0 \rangle\}$. Let σ_s denote the initial segment constructed before stage

s (it will be the case that $\text{content}(\sigma_s) = \{\langle e+1, x \rangle : x \in W_e^s\}$). Furthermore, $f_s(i)$ will be the function denoting counterexamples given to the learner M on its conjecture i (in the simulation at stage s). It will be the case that the range of f_s (except for $\#$) is a subset of E_s — which we will bar from belonging to $\{\langle e+1, x \rangle : x \in W_e\}$ to maintain the validity of any negative counterexample given. Initially, $f_0(i) = \#$, for all i , and $E_0 = \emptyset$. Go to stage 0.

Stage s

1. Simulate M by giving counterexamples according to f_s . Dovetail steps 2 and 3 until one of them succeeds. If step 2 succeeds before step 3, if ever, then go to step 4. If step 3 succeeds before step 2, if ever, then go to step 5. Here we assume that step 2 has some priority in the sense that if it can succeed for $t \leq s$, then it succeeds first, with σ being the shortest for which such $t \leq s$ exists.
2. Search for a $\sigma \subseteq \sigma_s$ and a t such that $W_{M(\sigma),t} - \text{content}(\sigma_s) \neq \emptyset$ and $\min(W_{M(\sigma),t} - \text{content}(\sigma_s)) \neq f_s(M(\sigma))$.
3. Search for a $\tau \supseteq \sigma_s$ such that $\text{content}(\tau) \subseteq \{\langle e+1, x \rangle : x \in \mathbb{N}\} - E_s$, and $M(\tau) \neq M(\sigma_s)$, where the negative counterexamples are given according to f_s .
4. Let

$$\begin{aligned} \sigma_{s+1} &= \sigma_s, \\ f_{s+1}(M(\sigma)) &= \min(W_{M(\sigma),t} - \text{content}(\sigma_s)), \\ f_{s+1}(i) &= f_s(i), \text{ for } i \neq M(\sigma), \\ W_e^{s+1} &= W_e^s. \\ E_{s+1} &= E_s \cup \{f_{s+1}(M(\sigma))\}, \text{ and} \end{aligned}$$

Go to stage $s+1$.

5. Let $\sigma_{s+1} = \tau$,
Let $W_e^{s+1} = \text{content}(\sigma_{s+1})$, $E_{s+1} = E_s$, $f_{s+1} = f_s$ and
Go to stage $s+1$

End stage s

Now, if there are infinitely many stages, then $W_e \in \mathcal{L}$, and $T = \bigcup_s \sigma_s$ is a text for W_e . As M identifies W_e , $M(T)$ converges. Then, for a large enough stage s , step 2 would not succeed anymore (as the least counterexamples would have been found by then). Thus, step 3 succeeds infinitely often, and M does not converge on T , a contradiction to the assumption that $M(T)$ converges.

Thus, there are only finitely many stages. Suppose stage s starts but does not end. Hence the counterexamples as in f_s on initial segments of σ_s are correct. Let y be such that $\langle 0, y \rangle \notin E_s$. Now consider the behaviour of M on $\sigma_s \langle 0, y \rangle^\infty$, where the counterexamples are the least counterexamples. If the learner does not converge, then it clearly does not identify $\sigma_s \langle 0, y \rangle^\infty$. Otherwise, let X be the set of counterexamples given, and let Y be the set of feedback queries asked on initial segments of the text $\sigma_s \langle 0, y \rangle^\infty$. Note that $X \cup Y$ is finite. Let x be such that $\langle e+1, x \rangle$ does not belong to $E_s \cup X \cup Y \cup \text{content}(\sigma_s)$. Then, M does not identify at least one of $\sigma_s \diamond \langle 0, y \rangle^\infty$ and $\sigma_s \diamond \langle e+1, x \rangle \diamond \langle 0, y \rangle^\infty$, as M 's conjectures converge to the same conjecture on both these texts. \square

Note that, obviously, the maximal element can always be memorized by a learner and, thus, cannot add more to the learning power of iterative learners than even one memory cell for storing input elements. Therefore, we explore if the number of elements seen so far can give an **NCIt**-learner more advantages than n memorized input elements seen so far. We were able to achieve only a partial solution — showing that the number of elements and the maximal element (or one memory cell) seen so far together can provide more power to **NCIt**-learners than n memorized input elements.

Theorem 24. *There exists a class \mathcal{L} such that*

- (a) \mathcal{L} can be **NCIt**-learnt using 1-memory (or the maximal element seen so far) and the number of elements seen so far, but
- (b) \mathcal{L} cannot be learnt by a **NCIt**-learner using n -feedback or n -memory, even if the learner is given the maximal element seen so far as additional input.

Proof. We say that e is *nice*, if $\emptyset \subset W_e \subseteq \{\langle e, j, x \rangle : j, x \in \mathbb{N}\}$, and for all x such that $\langle e, j, x \rangle \in W_e$, $W_e \cap \{\langle e, j, x' \rangle : j \in \mathbb{N}, x' < x\} \subseteq W_{e,x}$.

Now let $\mathcal{L}_1 = \{L : L = W_e, e \text{ is nice, and for all } x, \text{card}(W_e \cap \{j : \langle e, j, x \rangle \in L\} < x)\}$.

Let $\mathcal{L}_2 = \{L : (\exists e, x : x > 0)[W_e \in \mathcal{L}_1, W_e = W_{e,x} \text{ and } x > \max(W_e) \text{ and } \text{card}(\{j : \langle e, j, x \rangle \in L\}) = x \text{ and } L = W_{e,x} \cup \{\langle e, j, x \rangle : \langle e, j, x \rangle \in L\}]\}$.

Let $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$.

It is easy to see that \mathcal{L} can be iteratively learnt using the number of elements seen so far and 1-memory. The learner initially conjectures e (with padding) such that the first input element is $\langle e, j, x \rangle$ for some j, x . The learner remembers $\langle e, j, x \rangle$ for the largest x such that some element of the form $\langle e, j, x \rangle$ is in the input. Then, whenever the learner sees an input $\langle e, j', x' \rangle$ and the number of elements seen so far as k , it checks whether $\text{card}(W_{e, \max(\{x, x'\})}) + \max(\{x, x'\}) = k$. If so, then it proceeds to identify the input using the technique of Proposition 20.

\mathcal{L} cannot be **NCIt**-learnt using n -feedback or n -memory, as long as only the maximal element in the input is known. This can be shown essentially using the same technique as in the Theorems 15 and 16.

For n -feedback learning, in the diagonalization part, we do similar to Theorem 15, except that we do not need the part which was dealing with the “number of elements”. In the stage s , we choose a large enough $x > n + 3$, such that $W_{e,x}$ contains $\text{content}(\sigma_s)$. Then, we place $\langle e, n + 3 + j, x \rangle$ for $j < x - 2$ in W_e (instead of $\langle e, 0, m \rangle$ considered in the proof of Theorem 15). We let τ be an extension of σ_s such that $\text{content}(\tau) = \text{content}(\sigma_s) \cup \{\langle e, n + 3 + j, x \rangle : j < x - 2\}$. In step 4 we check if there exists a j such that M makes a mind change on $\tau \diamond \langle e, j, x \rangle$, where $1 \leq j \leq n + 2$. If so, then we proceed as in the construction in Theorem 15. Otherwise, we choose a j, j' such that $1 \leq j, j' \leq n + 2$, $j \neq j'$ and M on the previous conjecture $M(\tau)$ and the new element $\langle e, j, x \rangle$ does not query $\langle e, j', x \rangle$. Then, M fails to learn the input $\tau \diamond \langle e, j', x \rangle \langle e, j, x \rangle^\infty$.

Similar modification can be done for memory-bounded learning. \square

Can the maximal element give more power to **NCIt**-learners than the number of elements seen so far? The answer to this question is positive — even if the learners using the maximal element seen so far are just iterative (not using negative counterexamples to conjectures): this

is given by Theorem 22. However, we do not know whether the number of elements seen so far can give more in the context of **NCIt**-learnability than the maximal element.

Open Problem 25. *Is there a class \mathcal{L} of languages which can be **NCIt**-learnt using the number of elements seen so far as additional information (no memory or feedback) but cannot be **NCIt**-learnt using n -memory (or n -feedback with the maximal element seen so far as additional information or even just the maximal element seen so far as additional information)?*

Three following results give partial solutions to the open problems stated above. The first result shows that, under certain natural (and quite weak) assumptions, in the context of **NCIt**-learnability, access to the number of elements seen so far can be replaced by access to the maximal element seen so far.

Theorem 26. *Suppose \mathcal{L} can be **NCIt**-identified using the number of elements seen so far as additional information, where the learner is total (here the input element would be from the target language, but the number of elements may sometimes not be valid — we still expect the learner to converge). Then, \mathcal{L} can be **NCIt**-identified using the maximal element seen so far as additional information.*

Proof. Suppose M is an **NCIt**-learner using number of elements seen so far as additional information.

We construct M' as follows. Conjectures of M' will be of the form $P(\sigma, m, S, f)$, $Q(i, m, S, f)$, or $R(i, m, S, f)$, where $i, m \in \mathbb{N}$, $\sigma \in \text{SEQ}$, S is a finite set, and f is a finite function (giving counterexamples to conjectures). Furthermore, P, Q, R are 1–1 computable functions, with disjoint range, such that $W_{P(\sigma, m, S, f)} = W_{M(\sigma)}$ (where counterexamples are given according to f), $W_{Q(i, m, S, f)} = \{i\}$, and $W_{R(i, m, S, f)} = W_i$.

Intuitively, conjectures of the form $Q(\cdot, m, S, f)$ aim to find all elements $\leq m$ in the target language. Conjectures of the form $R(\cdot, m, S, f)$ find counterexamples, if any, to the conjectures $\leq m$. Conjectures of the form $P(\sigma, m, S, f)$ are for simulating M , as well as checking if σ seems like a stabilizing sequence for M on the target language, when counterexamples given are according to f .

The following properties will be satisfied by the construction:

(P1) From one conjecture to the next, the parameters m, S and f will be monotonically non-decreasing (where, for S, f this is in set-containment sense).

(P2) $S \subseteq L$. The domain of f is some initial segment (maybe empty) of \mathbb{N} . Furthermore, counterexamples, as given according to f for the target language, are correct for the conjectures in the domain of f .

(P3) Whenever $Q(i, m, S, f)$ is conjectured, S contains all the elements seen in the input upto that point, except, possibly, for elements between i and m (both inclusive). Furthermore, S also contains all the elements in the target language which are $< i$.

(P4) Whenever $R(i, m, S, f)$ is conjectured, S contains all the elements seen in the input upto that point. Also, S also contains all the elements in the target language which are $\leq m$. Furthermore, $0, 1, \dots, i - 1$, are in the domain of f .

(P5) When $P(\sigma, m, S, f)$ is first conjectured, S contains all the elements seen in the input upto that point; though later conjectures of the form $P(\sigma, m, S, f)$ (with same parameter values) may not satisfy this property. Furthermore, S contains all the elements in the target language which are $\leq m$, and f has domain at least $\{i : i \leq m\}$.

We now proceed to define M' formally as follows.

Initially, $M'(\lambda) = Q(0, 0, \emptyset, \emptyset)$.

(a) Suppose the previous conjecture is $Q(i, m, S, f)$, the new input is x and the counterexample is y .

(* Here we are trying to find membership in the target language for $z \leq m$, with the current round doing this for $z = i$ *).

If $y = \#$, then let $S = S \cup \{x, i\} - \{\#\}$; Else let $S = S \cup \{x\} - \{\#\}$.

If $i < m$, then conjecture $Q(i + 1, m, S, f)$; Else conjecture $R(i, m, S, f)$, where i is the least element not in the domain of f .

(b) Suppose the previous conjecture is $R(i, m, S, f)$, the new input is x , the counterexample is y , and m' is the maximal element seen so far.

(* Here we are trying to find counterexamples to conjectures $j \leq m$, with the current round doing so for $j = i$ *).

Let $f(i) = y$. Let $S = S \cup \{x\} - \{\#\}$.

If $i < m$, then conjecture $R(i + 1, m, S, f)$.

Else, if there exists a σ such that,

(i) $\text{content}(\sigma) \subseteq S \cap \{z : z \leq m\}$,

(ii) $|\sigma| \leq m$,

(iii) M (when given counterexamples according to f), on each initial segment of σ outputs only conjectures $\leq m$,

(iv) $M(\sigma \diamond \tau) = M(\sigma)$, for all τ such that $\text{content}(\tau) \subseteq S$ and $|\tau| \leq \text{card}(S) + 1$.

Then, conjecture $P(\sigma, m, S, f)$, for least such σ .

Else, conjecture $Q(0, m + m' + 1, S, f)$.

(c) Suppose the previous conjecture is $P(\sigma, m, S, f)$, the new input is x , the counterexample is y , and maximal element seen so far is m' .

Suppose $M(\sigma) = p$, where counterexamples are given according to f . Consider the output of M when the previous conjecture is p , the new input is x , and the number of elements seen so far is k , for $\text{card}(S \cup \{x\}) \leq k \leq \max(\{m, m'\}) - m + \text{card}(S \cap \{z : z \leq m\})$.

(* Intuitively, k takes all possible values for the number of elements seen so far which are possible for any set Z such that $S \subseteq Z \subseteq \{z : m < z \leq m' \text{ or } z \in S\}$. In particular, if $m > m'$, then $k = \text{card}(S)$ - this is useful if the target language is a finite set. *)

If all these conjectures of M are p , then M' continues with the conjecture $P(\sigma, m, S, f)$.

Else, M' conjectures $Q(0, m + 1 + m', S, f)$, where m' is the maximal element seen so far.

We now argue that if M -identifies the input language, then so does M' . Suppose L is the target language and T is the input text for L .

First note that, by induction, the invariants (P1) to (P5) are satisfied. This is easy to see for (P1) and (P2). For (P3) note that, whenever $Q(\cdot, \cdot, \cdot, \cdot)$ is conjectured in steps (b) or (c) above,

the value of m is at least as large as the largest element seen so far. Thus, (P3) also can be seen to hold by induction, as S is appropriately updated in step (a). Similarly, (P4) also holds, as conjecture $R(\cdot, \cdot, \cdot, \cdot)$ is made only in step (a) when the previous conjecture was of the form $Q(m, m, S, f)$, or in step (b), where by induction the property (P4) is maintained. Property (P5) holds, as the conjecture $P(\sigma, m, S, f)$ is made only via step (b) when $i = m$, or it is repeated via step (c).

Further, note that if a hypothesis of the form $Q(\cdot, m, \cdot, \cdot)$ is ever conjectured, then eventually, S would contain all elements of the target language which are $\leq m$ and all $i \leq m$ will be in the domain of f (via repeating conjectures of the form $Q(\cdot, m, \cdot, \cdot)$ and $R(\cdot, m, \cdot, \cdot)$ in steps (a) and (b)).

Also by construction, when a hypothesis of the form $P(\sigma, m, S, f)$ is first conjectured for a particular set of parameters (via step (b)), σ seems like a stabilizing sequence for inputs from S : in particular, $M(\sigma) = M(\sigma\tau)$, for all τ such that $\text{content}(\tau) \subseteq S$, and $|\tau| \leq \text{card}(S) + 1$ (we used '+1' to take care of \emptyset , as well as to handle the input $\#$); here the counterexamples to M are given via f , and the conjectures of M are all $\leq m$. Thus, if on the input text T , the learner M' converges to the hypothesis $P(\sigma, m, S, f)$, then $M(\sigma)$ must indeed be a grammar for the target language (as, for all elements in $L - S$, one checks in step (c) that M does not make a mind change, whatever the appropriate number of elements seen may be). Thus, M' also identifies the target language.

Also, either M' converges on T to some hypothesis of the form $P(\sigma, m, S, f)$ or a hypothesis of the form $Q(\cdot, \cdot, \cdot, \cdot)$ is conjectured infinitely often by M' on T (as step (b) eventually leads to a hypothesis of the form $P(\cdot, \cdot, \cdot, \cdot)$ or $Q(\cdot, \cdot, \cdot, \cdot)$, and step (c) eventually leads to hypothesis of the form $Q(\cdot, \cdot, \cdot, \cdot)$, unless the learner stabilizes on a hypothesis of the form $P(\cdot, \cdot, \cdot, \cdot)$). We will argue that the latter case (that is, a hypothesis of the form $Q(\cdot, \cdot, \cdot, \cdot)$ is conjectured infinitely often) cannot happen.

Note that in the latter case above, m grows unbounded (as both steps (b) and (c) increase the parameter m in the conjecture $Q(\cdot, m, \cdot, \cdot)$). Thus, the domain of f eventually contains every element, and S eventually contains all elements of L . Suppose σ is the least stabilizing sequence for M on L when counterexamples are given according to f . Let n be such that $\text{content}(\sigma) \subseteq L \cap \{x : x \leq n\}$, all conjectures of M on initial segments of σ are $\leq n$ (when counterexamples are given according to f), and in the case that L is finite, $\max(L) \leq n$. Furthermore, for all σ' with canonical index smaller than σ such that $\text{content}(\sigma') \subseteq L$, there is an $x \leq n$, $x \in L$ such that $M(\sigma') \neq M(\sigma'x)$ and all conjectures of M on initial segments of σ' are $\leq n$ (where the counterexamples given are the least ones). Then, once $Q(\cdot, m, \cdot, \cdot)$ is conjectured with $m \geq n$, we will have that eventually, $R(m, m, S, f)$ is conjectured for some parameter values S and f . Then, in step (b) (using invariants (P1) to P5), the stabilizing sequence σ would be chosen, and M' will conjecture $P(\sigma, m, S, f)$. This conjecture will then never be changed. \square

Our next theorem shows that, for iterative learners (not getting negative counterexamples to conjectures), the additional information regarding the number of elements seen so far can sometimes give advantage over n feedback queries and access to the maximal element seen so far. Note that the learner for the part (a) of the following theorem is not total.

Theorem 27. *There exists a class \mathcal{L} such that*

- (a) *\mathcal{L} can be iteratively learnt provided the learner is given the number of elements in the input seen so far as additional information.*
- (b) *For all n , \mathcal{L} cannot be iteratively learnt by an n -feedback learner even if it gets the maximal element seen so far as additional information.*

Proof. Let M_0, M_1, \dots denote a recursive enumeration of feedback query learners using maximal element seen so far, where the number of queries, per round, used by M_n is at most n .

Intuitively, for each n , we will place in the diagonalizing class some languages constructed specifically to diagonalize against M_n witnessing n -feedback learning using the maximal element seen so far as additional information. These languages will have n coded in every element of the language (so that a learner can obtain it easily). We will take care that the diagonalizing languages are never of cardinality $3s+1$. Furthermore, cardinality of the language being $\geq 3s+2$, implies that either the input language is of size $3s+2$ and contains some “special elements” or the construction of $\sigma_{n,s+1}$ below (forcing one extra mind change by M_n) was successful, or some terminal “diagonalizing condition” holds. The learner is able to determine which of the above cases holds, and thus output appropriately. We now proceed formally.

Using Operator recursion theorem [Cas74], there exists a recursive 1–1 increasing function p such that $W_{p(n)}$ may be defined as given below. Along with $W_{p(n)}$, we will also try to define $x_{n,0} < x_{n,1} < \dots$ as well as $\sigma_{n,0} \subseteq \sigma_{n,1} \subseteq \dots$. Always, $x_{n,s} = 1 + \max(\text{content}(\sigma_{n,s}))$. Note that $x_{n,s}$ is defined iff $\sigma_{n,s}$ is defined. $W_{p(n)} = \text{union of all } \text{content}(\sigma_{n,s}), \text{ where } \sigma_{n,s} \text{ gets defined.}$

Initially, $\sigma_{n,0} = \lambda$ and $x_{n,0} = 1$.

Definition of $\sigma_{n,s+1}$:

1. Wait until $M_n(\sigma_{n,s})$ and $M_n(\sigma_{n,s} \diamond \langle n, j, x_{n,s} \rangle)$ get defined for each $j \leq 2n+2$.
2. If $M_n(\sigma_{n,s} \diamond \langle n, j, x_{n,s} \rangle) \downarrow \neq M_n(\sigma_{n,s}) \downarrow$ for some j , Then
 Choose the least such j . Let j', j'' be least such that $j \neq j', j' \neq j'', j \neq j''$.
 Let $\sigma_{n,s+1} = \sigma_{n,s} \diamond \langle n, j, x_{n,s} \rangle \diamond \langle n, j', x_{n,s} \rangle \diamond \langle n, j'', x_{n,s} \rangle$.
 Let $x_{n,s+1} = 1 + \max(\text{content}(\sigma_{n,s+1}))$.
 Else Loop forever ($\sigma_{n,s+1}$ does not get defined in this case).

Note that $\text{card}(\sigma_{n,s}) = 3s$.

The class \mathcal{L} will consist of the following languages (for each n),

- (i) $W_{p(n)}$,
- (ii) the languages $\text{content}(\sigma_{n,s}) \cup \{\langle n, j, x_{n,s} \rangle, \langle n, 2n+3+j, x_{n,s} \rangle\}$, for $j \leq 2n+2$, whenever $\sigma_{n,s}$ is defined.
- (iii) In case $\sigma_{n,s}$ is defined and $M_n(\sigma_{n,s} \diamond \langle n, j, x_{n,s} \rangle) \downarrow = M_n(\sigma_{n,s}) \downarrow$, for all $j \leq 2n+2$, then \mathcal{L} will additionally contain $\text{content}(\sigma_{n,s}) \cup \{\langle n, j_1, x_{n,s} \rangle, \langle n, 2n+2, x_{n,s} \rangle\}$ and $\text{content}(\sigma_{n,s}) \cup \{\langle n, j_1, x_{n,s} \rangle, \langle n, j_2, x_{n,s} \rangle, \langle n, 2n+2, x_{n,s} \rangle\}$, where
 $j_1 < 2n+2$ is maximal such that M_n on previous conjecture $M(\sigma_{n,s})$ and new input $(\langle n, 2n+2, x_{n,s} \rangle)$ does not query $\langle n, j_1, x_{n,s} \rangle$ and

$j_2 < j_1$ is maximal such that M_n on previous conjecture ($M(\sigma_{n,s})$ and new input $\langle n, 2n+2, x_{n,s} \rangle$) or $\langle n, j_1, x_{n,s} \rangle$ does not query $\langle n, j_2, x_{n,s} \rangle$ (note that in this case, $\sigma_{n,s+1}$ does not get defined, and $W_{p(n)} = \text{content}(\sigma_{n,s})$).

Claim 28. \mathcal{L} can be iteratively learnt using the number of elements in the input seen so far as additional information.

Proof. An iterative learner (with the number of elements in the input seen so far as additional information) can learn \mathcal{L} by initially outputting a grammar for \emptyset . It can determine $p(n)$ when it first sees $\langle n, \cdot, \cdot \rangle$. Beyond (and including) this point, if it ever sees an element of the form $\langle n, j, x \rangle$, where $j > 2n+2$, it will make the conjecture $\text{content}(\sigma_{n,s}) \cup \{\langle n, j, x \rangle, \langle n, j-2n-3, x \rangle\}$ (where s is such that $x = x_{n,s}$), as in the clause (ii) in the definition of \mathcal{L} and never change its mind thereafter.

Otherwise, if the number of input elements seen so far is $3s+1$, for some s , then the learner outputs $p(n)$. If the number of elements seen so far is $3s+2$ or $3s+3$, then it finds $x_{n,s}$ and $\sigma_{n,s}$ (which has $3s$ number of elements). It then waits until $M_n(\sigma_{n,s})$ and $M_n(\sigma_{n,s} \diamond \langle n, j, x_{n,s} \rangle)$ get defined for each $j \leq 2n+2$. If there exists a $j \leq 2n+2$, such that $M_n(\sigma_{n,s} \diamond \langle n, j, x_{n,s} \rangle) \neq M_n(\sigma_{n,s})$, then it continues to output $p(n)$. Otherwise, it determines the maximal j_1 such that M_n on the previous conjecture $M_n(\sigma_{n,s})$ and new input $\langle n, 2n+2, x_{n,s} \rangle$ does not query $\langle n, j_1, x_{n,s} \rangle$ and maximal $j_2 < j_1$ such that M_n on the previous conjecture $M_n(\sigma_{n,s})$ and new input being $\langle n, 2n+2, x_{n,s} \rangle$ or $\langle n, j_1, x_{n,s} \rangle$ does not query $\langle n, j_2, x_{n,s} \rangle$. The learner then outputs a grammar for $\text{content}(\sigma_{n,s}) \cup \{\langle n, j_1, x_{n,s} \rangle, \langle n, j_2, x_{n,s} \rangle, \langle n, 2n+2, x_{n,s} \rangle\}$, if the number of elements seen so far was $3s+3$; the learner outputs a grammar for $\text{content}(\sigma_{n,s}) \cup \{\langle n, j_1, x_{n,s} \rangle, \langle n, 2n+2, x_{n,s} \rangle\}$, if the number of elements seen so far is $3s+2$. It is easy to verify that the above learner will iteratively learn \mathcal{L} using the number of elements seen as additional information. This completes the proof of the claim.

Claim 29. M_n cannot iteratively learn \mathcal{L} using n -feedback queries, even if it is given the maximal element seen so far as additional information.

Proof. To see this, suppose infinitely many $\sigma_{n,s}$ get defined. Then clearly M_n does not **TextIt** learn $W_{p(n)}$ using n -feedback from the text $\bigcup_s \sigma_{n,s}$, as there are infinitely many mind changes by M_n on the text.

On the other hand, if $\sigma_{n,s+1}$ does not get defined, then if step 1 does not finish, then M_n does not learn the language $\text{content}(\sigma_{n,s}) \cup \{\langle n, j, x_{n,s} \rangle, \langle n, 2n+3+j, x_{n,s} \rangle\}$ for the j for which $M_n(\sigma_{n,s} \diamond \langle n, j, x_{n,s} \rangle)$ does not converge.

If step 2 does not finish, then $M_n(\sigma_{n,s} \diamond \langle n, j, x_{n,s} \rangle) = M_n(\sigma_{n,s})$, for each $j \leq 2n+2$. Let $j_1 < 2n+2$ be maximal such that M_n on the previous conjecture $M_n(\sigma_{n,s})$ and the new input $\langle n, 2n+2, x_{n,s} \rangle$ does not query $\langle n, j_1, x_{n,s} \rangle$ and let $j_2 < j_1$ be maximal such that M_n on the previous conjecture $M_n(\sigma_{n,s})$ and the new input being $\langle n, 2n+2, x_{n,s} \rangle$ or $\langle n, j_1, x_{n,s} \rangle$ does not query $\langle n, j_2, x_{n,s} \rangle$. Then, M_n fails to identify at least one of $\text{content}(\sigma_{n,s}) \cup \{\langle n, j_1, x_{n,s} \rangle, \langle n, 2n+2, x_{n,s} \rangle\}$ and $\text{content}(\sigma_{n,s}) \cup \{\langle n, j_1, x_{n,s} \rangle, \langle n, j_2, x_{n,s} \rangle, \langle n, 2n+2, x_{n,s} \rangle\}$, as M_n converges to the

same grammar on both $\sigma_{n,s} \diamond \langle n, j_1, x_{n,s} \rangle \diamond \langle n, 2n+2, x_{n,s} \rangle^\infty$ and $\sigma_{n,s} \diamond \langle n, j_2, x_{n,s} \rangle \diamond \langle n, j_1, x_{n,s} \rangle \diamond \langle n, 2n+2, x_{n,s} \rangle^\infty$. \square

A similar idea can be used to show that, for iterative learners, the number of elements seen so far can sometimes give more advantage than n stored elements seen of the input.

Note that the learner in part (a) of the following theorem is not total.

Theorem 30. *There exists a class \mathcal{L} such that*

(a) *\mathcal{L} can be iteratively learnt provided the learner is given the number of elements seen so far as additional information.*

(b) *For all n , \mathcal{L} cannot be iteratively learnt by an n -memory bounded learner.*

6 Using the Length of the Input as Additional Information

The length of the input seen so far can potentially be viewed as an alternative to the number of elements seen so far as a source of additional information for **NCIt**-learners. However, we show in this section that, for **NCIt**-learners, it can be replaced by access to the maximal element seen so far. The technique used for the following theorem is similar to that used in the proof of Theorem 26, with some modifications to handle length.

Theorem 31. *Suppose \mathcal{L} is **NCIt**-learnable by a learner using the length of input as additional information. Then \mathcal{L} is **NCIt**-learnable by a learner using the maximal element seen so far as additional information.*

Proof. Suppose M is an **NCIt** learner which uses the length of input. We will define M' below which uses the maximal element seen so far and **NCIt**-identifies the languages which are **NCIt**-learnt by M .

Below f will denote a finite partial function such that $f(i)$ is a counterexample (or $\#$) to conjecture i . The defined part of f will always be correct with respect to the input language L .

Let P be a 1–1 recursive function such that $W_{P(\sigma,f)} = W_{M(\sigma)}$, if $M(\sigma) = M(\sigma\#^k)$ for all k (where counterexamples are provided according to f), and $f(M(\sigma'))$ is defined for all prefixes σ' of σ ; Otherwise $W_{P(\sigma,f)} = \mathbb{N}$ (this condition applies if either $f(M(\sigma'))$ is not defined for some prefix σ' of σ , or $M(\sigma) \neq M(\sigma\#^k)$ for some k).

We now define M' . M' would be an **NCIt**-learner using the maximal element seen so far as additional information. For ease of presentation, we give M' repeatedly requesting inputs, the maximal element seen so far, and making conjectures and receiving counterexamples. Initially, M' outputs a conjecture for \mathbb{N} (which we assume without loss of generality not to be syntactically same as any of the other conjectures used below). If there is no counterexample, then we are done. Otherwise M' goes to stage 0 below. Note that M' can remember f , stage number, phase and which part of the phase (in case of phase 1) it is in, by just padding its conjecture appropriately; Thus, for phase 1, we essentially describe it as if M' can remember all the data it has seen since the phase started — in phase 1, we are not concerned about converging to a hypothesis, but just about finding certain σ_s — thus all new input seen can be padded.

Stage s

Phase 1:

1. Suppose m is the maximal element seen so far. Below τ will denote the sequence of elements seen in the input since this phase started (note that this τ will keep getting updated with time, based on which step we are executing).
2. Let Y denote the set of elements $\leq m$ which belong to the input language (this can be determined using (padded) conjectures for $\{x\}$, with $x \leq m$). Note here that each such conjecture would update τ too.
3. Loop for $t = 0$ to ∞ :
 - 3.1 Update f by finding the value of f on the least number on which it is not defined. This can be done by conjecturing the least input on which f is not defined (τ correspondingly gets updated).
Suppose $t = \langle t', t'' \rangle$. Let σ_s be the sequence with the canonical code t' .
Below, in the simulation of M , counterexamples are provided using f .
If
 - 3.2 $\text{content}(\sigma_s) \subseteq Y \cup \text{content}(\tau)$, $f(M(\sigma'))$ is defined for each prefix σ' of σ_s and
 - 3.3 $f(P(\sigma_s, f')) = \#$, where f' is restriction of f to the domain being $\{M(\sigma') : \sigma' \subseteq \sigma_s\}$, and
 - 3.4 $M(\sigma_s \#^x \cdot x) = M(\sigma_s)$, for all $x \in Y \cup \text{content}(\tau)$.
 Then go to step 4. Otherwise go to the next iteration of the loop.
 - End Loop
 4. If appropriate σ_s is found, then M' outputs (padded) $P(\sigma_s, f)$ and goes to Phase 2.

Phase 2

On the new input x check if $M(\sigma_s) = M(\sigma_s \#^x x)$. If not, then go to stage $s+1$. Otherwise repeat $P(\sigma_s, f)$ as the conjecture.

End stage s

Now we claim that the above M' **NCIt**-identifies (using the maximal element seen so far as additional information) any language which is **NCIt**-learnt by M (using the length of input as additional information).

Let L be the input language. Values of variables below are as at the corresponding stage/phase. If the input language is \mathbb{N} , then clearly M' -identifies it.

Now suppose the above learner gets stuck in phase 2 of some stage s . Let T' be a text for L where $T'(x) = x$, if $x \in L$, and $T'(x) = \#$ otherwise. Then clearly, when counterexamples are given according to f as at the beginning of phase 2 in stage s , for all n , $M(\sigma_s T'[n]) = M(\sigma_s)$ (otherwise either $P(\sigma_s, f)$ would have generated a counterexample, or a mind change would have been found on $M(\sigma_s \#^x x)$ for some $x \in Y \cup \text{content}(\tau)$ as at step 3.4 of phase 1, or when input x is received in phase 2; here note that Y contains all the data seen in stages before stage s and

($M(\sigma_s) = M(\sigma_s \#^x)$ and $M(\sigma_s) = M(\sigma_s T'[x])$) implies $M(\sigma_s \#^x x) = M(\sigma_s T'[x+1])$. Moreover, all the answers given by f are correct. Thus, M' **NCIt**-identifies L (using the maximal element seen so far as additional information).

If M' gets stuck at phase 1 of some stage s , then there is no stabilizing sequence for M on input L (when the counterexamples are given according to f), and thus M does not **NCIt**-learn L . (Here note that Y contains all the data seen by M' in stages before stage s , and thus all elements of the input language are eventually in $Y \cup \text{content}(\tau)$, as step 3.1 gets executed infinitely often; moreover f gets defined on all inputs).

We now argue that there cannot be infinitely many stages. Note that if there are infinitely many stages, then f eventually gets defined on all inputs — for the computation of M below, we assume that the counterexamples are given according to this f . Let $\langle t', t'' \rangle$ be the least such that σ with the canonical index t' satisfies the conditions: (i) $\text{content}(\sigma) \subseteq L$, (ii) for all $x \in L$, $M(\sigma \#^x x) = M(\sigma)$ and $M(\sigma \#^k) = M(\sigma)$ for all k . Note that there exists such a σ , as every stabilizing sequence satisfies these properties.

Let s be large enough so that (i) input text $T[s]$ contains all elements in $\text{content}(\sigma)$, (ii) for all $\langle t'_1, t''_1 \rangle < \langle t', t'' \rangle$, for γ with the canonical index t'_1 , either $\text{content}(\gamma) \not\subseteq \text{content}(T)$, or $M(\gamma) \neq M(\gamma \#^x x)$ for some $x \in \text{content}(T[s])$, or $M(\gamma \#^k) \neq M(\gamma)$, for some k . (Here, note that after the execution of step 2 at stage s at least the first s elements of the input text are already in Y .) Now, in the loop at step 3 in stage s , Phase 1, σ_s would be defined to be σ , and thus the learner will not leave stage s (phase 2) anymore. \square

7 Robustness of NCIt-learning with Additional Information

In this section we consider whether our results still hold if instead of giving the maximal/number of elements seen so far as additional information the learner is only given an upper bound on these values or an approximate value which is within an additive constant c of the actual value. This would in some sense show robustness of the results against error in the additional information. Below we briefly sketch how each of the results relating to the maximal element/number of elements seen so far is affected when one considers such a modification in the additional information.

Note that an upper bound on the maximal element seen so far implies an upper bound on the number of elements seen so far (which means that a bound on the number of elements seen so far can always be simulated using a bound on the maximal element).

First we note that the maximal element seen so far can be replaced by an upper bound on the maximal element seen so far in the proof of Theorem 21.

The next question is whether Theorem 22, which shows the strength of **NCIt**-learners using the maximal element seen so far against the ones using the number of elements seen so far, can be extended to the case when only an upper bound for the maximal element seen so far is available to the learner. We don't know the answer to this question.

Theorem 24 shows the strength of the maximal element seen so far (or 1-cell memory) and the number of elements seen so far available to **NCIt**-learners together. The proof of this theorem

does not work if, on the positive side, the learner is given an upper bound on the number of elements seen so far, or an approximation to the number of elements seen so far within an additive constant. However, one can modify the proof of Theorem 24 to work for the case when the positive side is given an approximation to the upper bound on the number of elements seen so far (within an additive constant c), by replacing $L \in \mathcal{L}$ by L' , where

$$x \in L \text{ iff } L' \text{ contains } (6c + 3)x + 2c + 2 + j, \text{ for } 0 \leq j < 2c + 1.$$

The idea here is that each x is mapped to a group of $6c + 3$ elements, where the least and the highest $2c + 1$ elements are not in the language L' , and the middle $2c + 1$ ones are in L' iff x is in L ; this essentially allows a learner, given an approximation within a “constant c ” for the number of elements in the language (in case it is finite), to compute the actual number of elements in the input language. This is enough for learning the class \mathcal{L} as in Theorem 24. The diagonalization proof as in Theorem 24 also can be adjusted appropriately, as the proof there worked for all possible n -feedback and n -memory learners. The same idea can be used for similar cases below.

Also note that Theorem 24 does not work if the upper bound on the number of elements seen so far is given on the positive side, as the maximal element seen so far also bounds the number of elements seen so far too. So the diagonalization against n -memory bounded **NCIt**-learner does not work in this case.

For Theorem 23, the proof does not work if we are given an upper bound on the number of elements seen so far (the proof does work when we are given an upper bound on the maximal element seen so far). However, the proof can be modified to work for the case when an upper bound on the number of elements seen so far is given to the learner. This is done by first partitioning \mathbb{N} into blocks I_0, I_1, I_2, \dots , where I_k is of size $k + 1$. The languages in \mathcal{L}_2 , are of the form $\{\langle e, x \rangle : x \in I_k, k \in D\} \cup \{\langle 0, y \rangle\}$, for some finite D . Thus, essentially, the upper bound on the number of elements seen so far gives away the upper bound on the maximal element for the languages in \mathcal{L}_2 . This allows one to identify the class \mathcal{L} . The diagonalization proof can easily be modified to use the updated class.

Theorem 26 shows that, for **NCIt**-learners converging on all inputs, the number of elements so far can be simulated by the maximal element seen so far. This simulation holds if only an upper bound on the maximal element seen so far is given.

For Theorem 27, showing advantage of iterative learners using access to the number of elements seen so far over the ones using n -feedback and the maximal element seen so far, the proof can be modified to handle the case when, on the positive side, the learner is given an approximation to the number of elements seen so far within an additive constant c (by using $2c + 1$ cylinderfication). The proof, of course, cannot be modified to work for an upper bound on the number of elements seen so far, as it is bounded by the upper bound on the maximal element seen so far.

8 Conclusions

As we have shown, additional information of the types studied in this paper can add interesting new capabilities to iterative learners getting negative examples to conjectures containing

data in excess of the target language. Some problems related to comparisons of help provided by additional information remain open, and solving these problems can offer new (and, possibly, unexpected) insight into advantages of using additional information of certain types for the learners in question. Influence of noise on additional information has been discussed in Section 7, however, many questions remain open here as well. Similarly to [JK07], one might also consider different types of negative examples (refuting conjectures containing extra elements) by iterative learners and explore how these different types of negative examples may interplay with different types of additional information. Yet another interesting area of research is studying iterative learnability with counterexamples and additional information of specific indexed classes of languages (for example, regular languages or patterns) — as we have shown all such classes are learnable class-preservingly using the maximal element or the number of elements seen so far as additional information, and, therefore, one can now study if and when learnability of such classes may be efficient.

A general open problem for iterative learners of any type using additional (bounded) memory is whether a *multiset* type memory (when a learner can store the same input item several times; for example, the learner may decide to store, say, 10 copies of the next input element) can have an advantage over a *set* type memory (where every item is stored just once). We suspect that no such advantage is possible — however, we have not been able to find a proof.

Acknowledgments. A preliminary version of the paper appeared in [JK09]. The authors are grateful to the anonymous referees of ALT’2009 and this journal for many useful remarks and suggestions. We specially thank a referee for a simpler proof of Theorem 10.

References

- [Ang80] D. Angluin. Finding patterns common to a set of strings. *Journal of Computer and System Sciences*, 21(1):46–62, 1980.
- [Ang88] D. Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.
- [BA96] R. Brachman and T. Anand. The process of knowledge discovery in databases: A human centered approach. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusam, editors, *Advances in Knowledge Discovery and Data Mining*, pages 37–58. AAAI Press, 1996.
- [BB75] L. Blum and M. Blum. Toward a mathematical theory of inductive inference. *Information and Control*, 28(2):125–155, 1975.
- [Blu67] M. Blum. A machine-independent theory of the complexity of recursive functions. *Journal of the ACM*, 14(2):322–336, 1967.
- [Cas74] J. Case. Periodicity in generations of automata. *Mathematical Systems Theory*, 8(1):15–32, 1974.
- [CJLZ99] J. Case, S. Jain, S. Lange, and T. Zeugmann. Incremental concept learning for bounded data mining. *Information and Computation*, 152(1):74–110, July 1999.

- [CL82] J. Case and C. 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*, volume 140 of *Lecture Notes in Computer Science*, pages 107–115. Springer-Verlag, 1982.
- [CM08] J. Case and S. Moelius. U-shaped, iterative, and iterative-with-counter learning. *Machine Learning*, 72(1–2):63–88, 2008.
- [FPSS96] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusam, editors, *Advances in Knowledge Discovery and Data Mining*, pages 1–34. AAAI Press, 1996.
- [Ful90] M. Fulk. Prudence and other conditions on formal language learning. *Information and Computation*, 85(1):1–11, 1990.
- [Gol67] E. M. Gold. Language identification in the limit. *Information and Control*, 10(5):447–474, 1967.
- [HU79] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.
- [JK07] S. Jain and E. Kinber. Iterative learning from positive data and negative counterexamples. *Information and Computation*, 205(12):1777–1805, 2007.
- [JK08] S. Jain and E. Kinber. Learning languages from positive data and negative counterexamples. *Journal of Computer and System Sciences*, 74(4):431–456, 2008. Special Issue: Carl Smith memorial issue.
- [JK09] S. Jain and E. Kinber. Iterative learning from texts and counterexamples using additional information. In R. Gavalda, G. Lugosi, T. Zeugmann, and S. Zilles, editors, *Algorithmic Learning Theory: 20th International Conference (ALT’ 2009)*, volume 5809 of *Lecture Notes in Artificial Intelligence*, pages 308–322. Springer-Verlag, 2009.
- [Joc68] C. G. Jockusch. Semirecursive sets and positive reducibility. *Transactions of the American Mathematical Society*, 131:420–436, 1968.
- [LZ92] S. Lange and T. Zeugmann. Types of monotonic language learning and their characterization. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 377–390. ACM Press, 1992.
- [LZ96] S. Lange and T. Zeugmann. Incremental learning from positive data. *Journal of Computer and System Sciences*, 53(1):88–103, 1996.
- [LZ06] Y. Li and W. Zhang. Simplify support vector machines by iterative learning. *Neural Processing Information - Letters and Reviews*, 10(1):11–17, 2006.
- [LZZ08] S. Lange, T. Zeugmann, and S. Zilles. Learning indexed families of recursive languages from positive data: A survey. *Theoretical Computer Science*, 397(1–3):194–232, 2008.
- [OSW86] D. Osherson, M. Stob, and S. Weinstein. *Systems that Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists*. MIT Press, 1986.
- [Pop68] K. Popper. *The Logic of Scientific Discovery*. Harper Torch Books, New York, second edition, 1968.
- [Rog67] H. Rogers. *Theory of Recursive Functions and Effective Computability*. McGraw-Hill, 1967. Reprinted by MIT Press in 1987.

- [Wie76] R. Wiehagen. Limes-Erkennung rekursiver Funktionen durch spezielle Strategien. *Journal of Information Processing and Cybernetics (EIK)*, 12(1–2):93–99, 1976.